



Introduction to HIPE

Bernhard Schulz
NHSC/IPAC





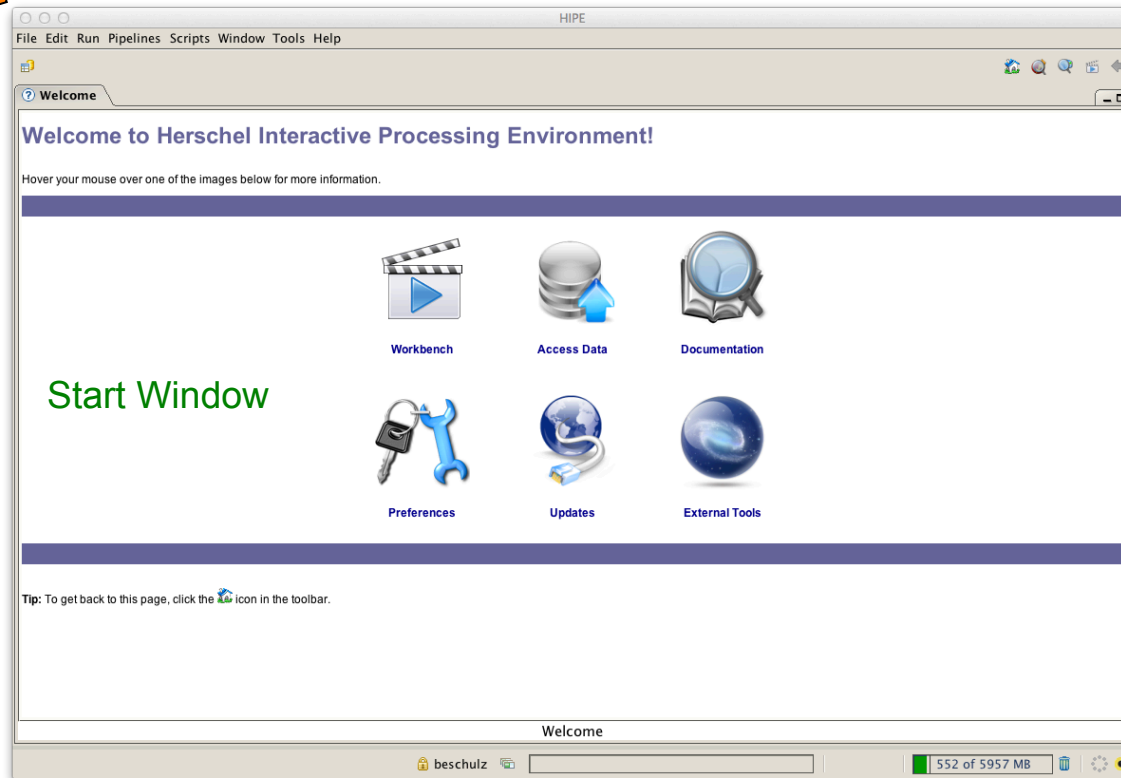
What is HIPE

- HIPE (Herschel Interactive Processing Environment)
 - GUI front end to the Herschel Common Science System (HCSS) for data processing.
- Written in Java
 - supported under Linux, Mac OS, MS Windows
- The software was developed in parallel to the mission hardware.
- All standard data products in the Herschel Archive were processed with this software.
- The code is free, open, and the same software is available to astronomers to repeat and improve Herschel data processing.
- HIPE was built to make the access to the data and data reduction procedures more user-friendly.
- Supports Jython as user scripting language.
- This presentation is here to whet your appetite and give you a basic introduction and overview of its capabilities.



First Look at HIPE

Warning:
Closing the
window will
terminate the
session
immediately.



Archive-
connected
indicator



Progress bar



Memory bar



SAMP-
connected
indicator



Busy
indicator





Getting Help

The image displays two overlapping windows from the HIPE (Herschel Interactive Processing Environment) software. The left window is the main application interface, titled 'Welcome to Herschel Interactive Processing'. It features a menu bar with 'File', 'Edit', 'Run', 'Pipelines', 'Scripts', 'Window', 'Tools', and 'Help'. Below the menu is a 'Welcome' tab and a large orange arrow pointing to the 'Help' menu item. The main content area has a clapperboard icon labeled 'Workbench' and a wrench/key icon labeled 'Preferences'. A tip at the bottom says: 'Tip: To get back to this page, click the icon in the toolbar.' The right window is the 'HIPE - welcome - Welcome to the HIPE Help System'. It has a search bar and a table of contents (TOC) on the left. The TOC lists sections: 'Introductory', 'Analysis Tools', 'HIFI', 'SPIRE', 'PACS', 'Reference', and 'Developer Reference'. The main content area is titled 'Welcome to the Herschel Interactive Processing Environment Help System' and contains a grid of help topics, each with an icon and a link: 'New to HIPE? Click here for a quick introduction', 'Learn about the Help System: Help on Help', 'Learn how to get data from the Herschel Archive', 'Learn about HIFI data: Data reduction guide, Pipeline specification, Data known issues', 'Learn about PACS data: Data reduction guide for photometry and spectroscopy, Data known issues', 'Learn about SPIRE data: Data reduction guide, Pipeline specification, Data known issues', and 'HIPE known issues'.



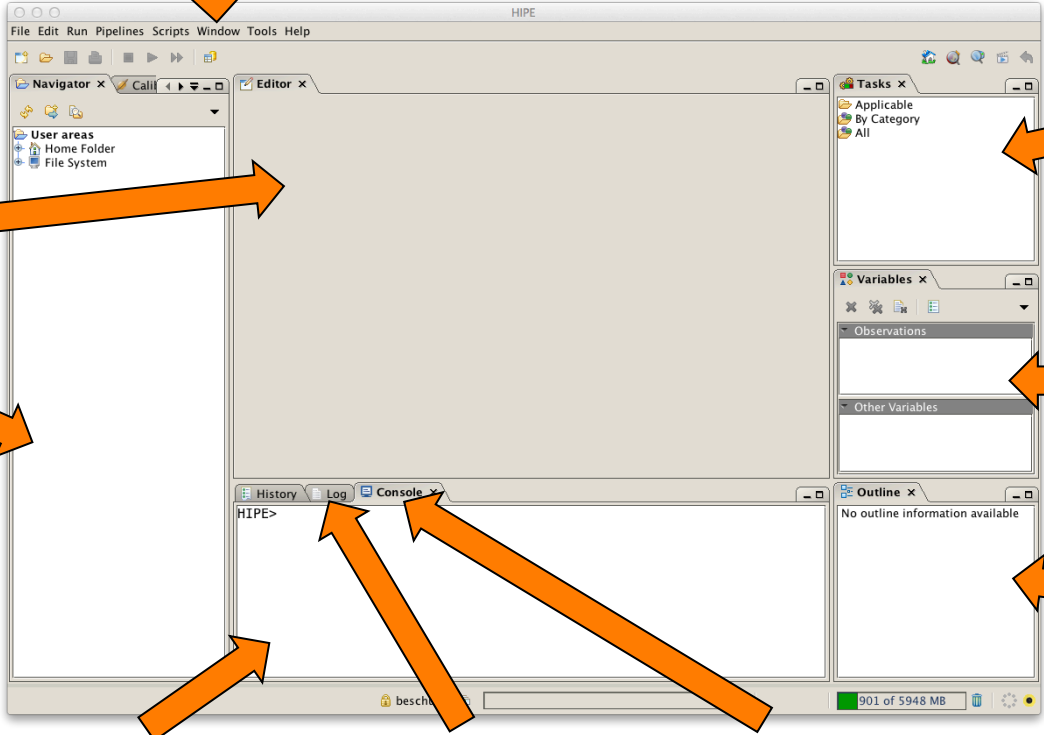


HIPE GUI Components (Views)

more here

Editor Window
Jython procedure editing
Data viewers

Navigator
Window to file system
Import data tarballs



Tasks View
Show available and applicable Tasks

Variables View
Shows variables of the session. Observations are shown in a special section.

Outline View
Shows internal structure of selected variable.

Console
Command line interface to Jython.

History View
Records command history and provides error information.

Log View
Logs output from tasks and procedures.

There are a few more views to discover...



Perspectives

- Views can be arranged in many different ways, beside each other or stacked in tabs using **drag-and-drop**.
- Such an arrangement is called “**Perspective**”
- Three different **pre-set perspectives** are available by clicking the icons on the upper right of the panel.
- Any changes to the current perspective are remembered automatically in the last selected pre-set.
- The pre-sets can be **reset to their original state**.

start window

pre-sets

pre-set reset

Views can be stacked in tabs behind each other

drag and drop views by their tab header to another position

drag borders to change size

```
1 from java.lang.Math import PI
2
3 lst = ['Jupiter', PI, True]
4
5 for item in lst:
6     print item
7
8
```

HIPE> from java.lang.Math import PI
HIPE> lst = ['Jupiter', PI, True]
HIPE> for item in lst:
 print item
Jupiter
3.14159265359
True
HIPE>

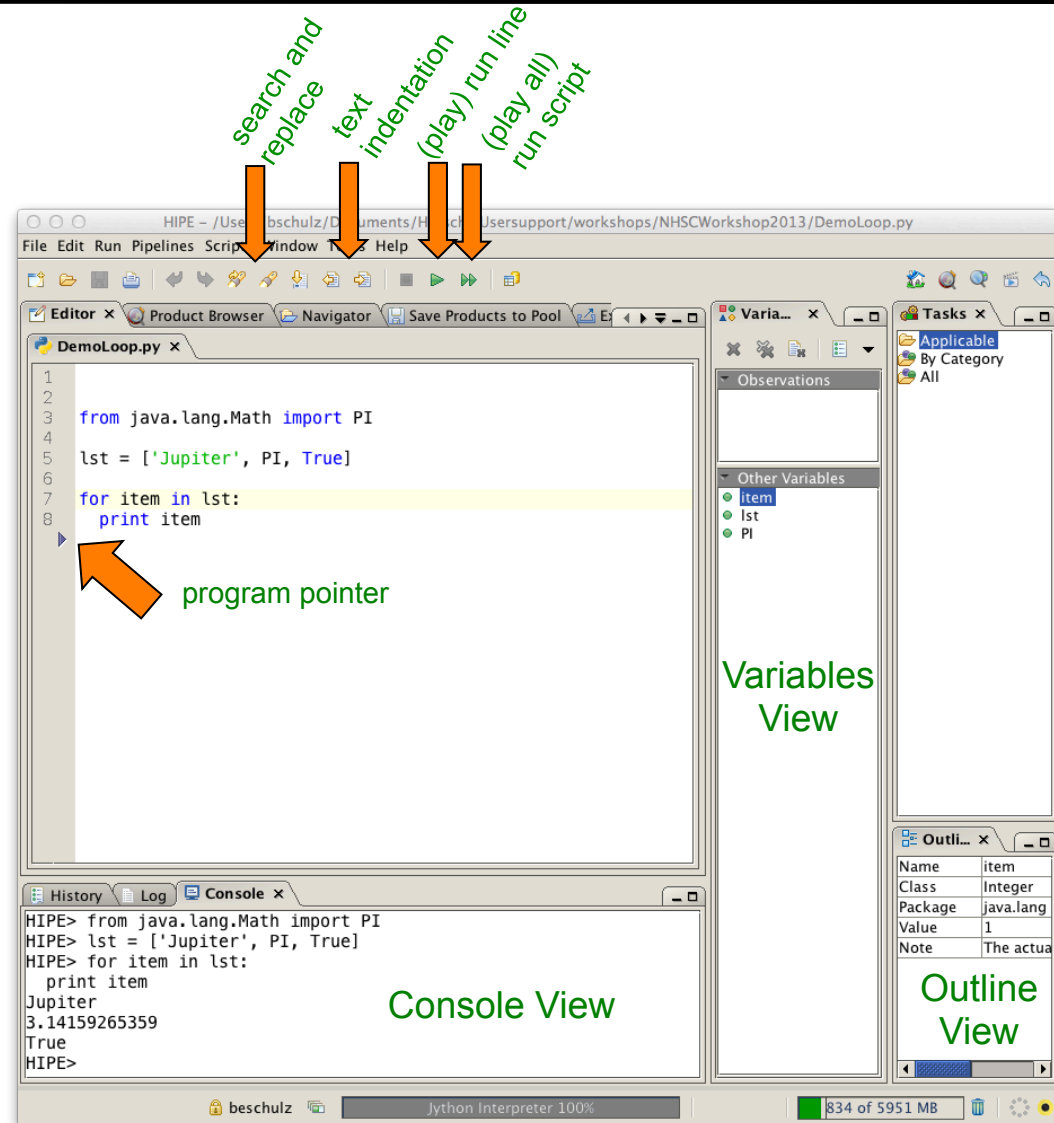
Name	item
Class	Integer
Package	java.lang
Value	1
Note	The actual objec

715 of 5951 MB



Editor and Console Views

- The Console View works like a **Python command line interface**.
- Program code in the editor window can be executed using the **green “play” arrows** in the toolbar on the top.
- Lines selected by the small triangle (**program pointer**) to the left or highlighted code will be executed by clicking the single green arrow.
- The **entire file** will be executed clicking the green double arrow.
- More editor functions exist like search/replace, and shift text blocks right or left.
- Variables and Outline Views are updated when code is executed.





System Components

- HIPE GUI front end
- Jython interpreter
- Libraries (Numerics, Plot, IO, etc.)
- Pipeline and User scripts
- File system
- Databases (Pools, HSA)



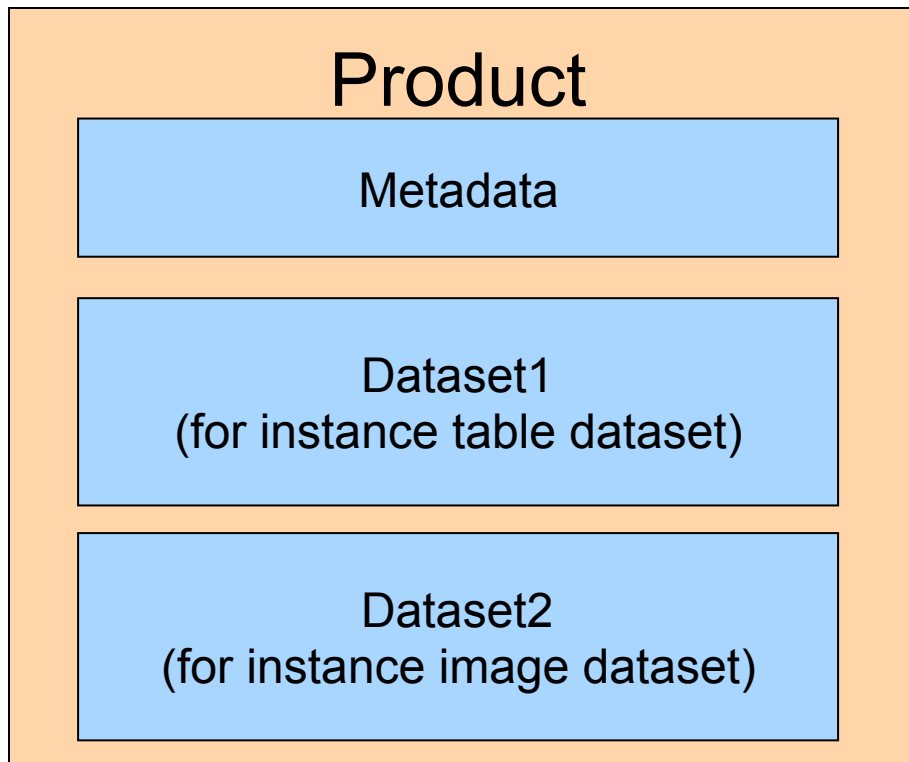
Products

- In HIPE data is held within containers called products.
- These units were designed with the FITS file format in mind.
- A product can be exported from HIPE and saved as a FITS file.
- Datasets are saved by encapsulating them into a product.
- Every product has metadata.



General Product Structure

- Products are containers for datasets that can be stored within the HCSS system.
- A product can be stored in a pool or exported to a FITS file .

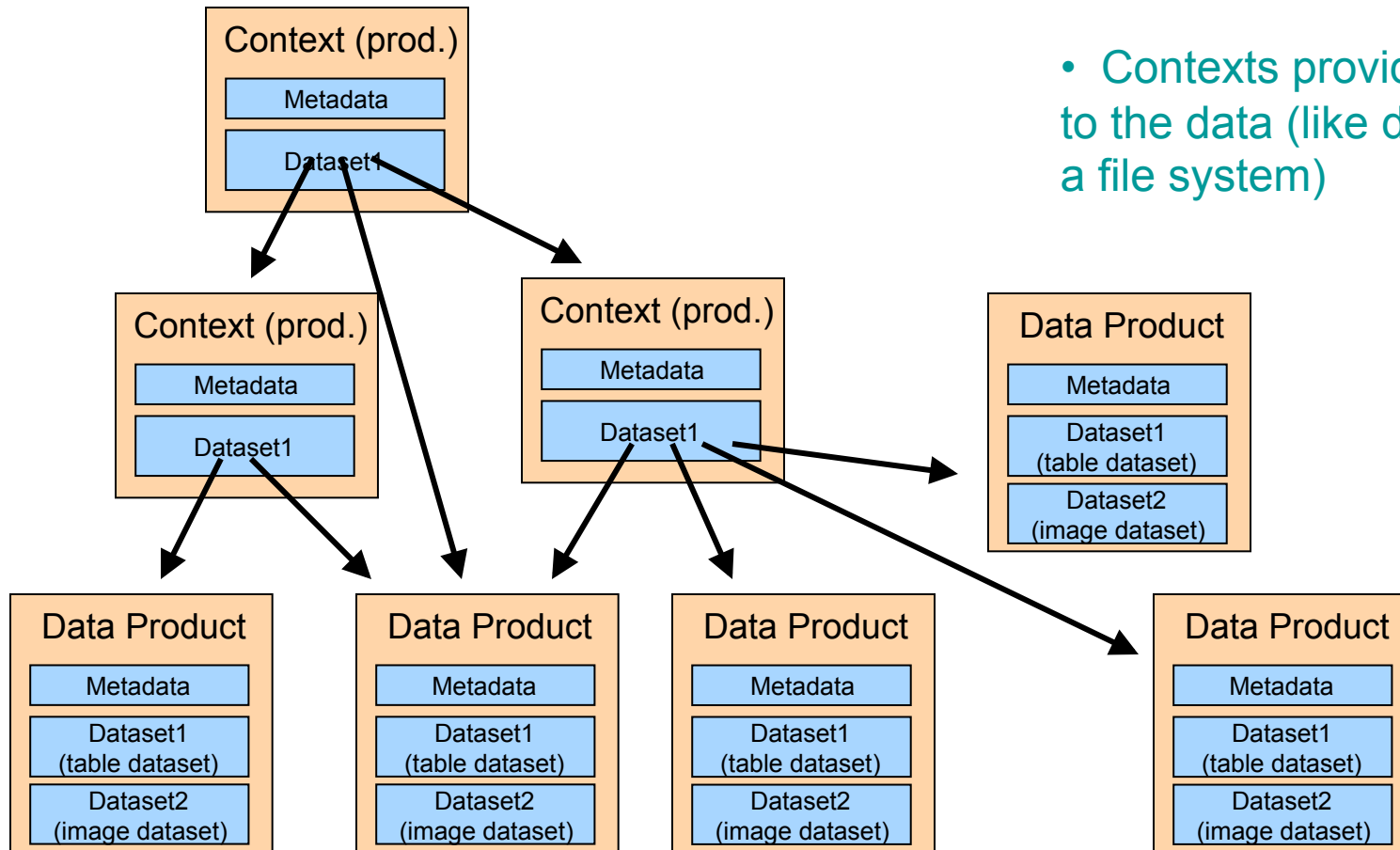


- Products contain:
 - Metadata,
 - Datasets
 - Processing history
- Types of datasets are:
 - Array dataset
 - Table dataset
 - Composite dataset
 - Spectrum1d
 - Spectrum2d
- Generic Product Types are:
 - SimpleImage
 - SimpleCube
 - SpectralSimpleCube
 - **Context** ←



Contexts

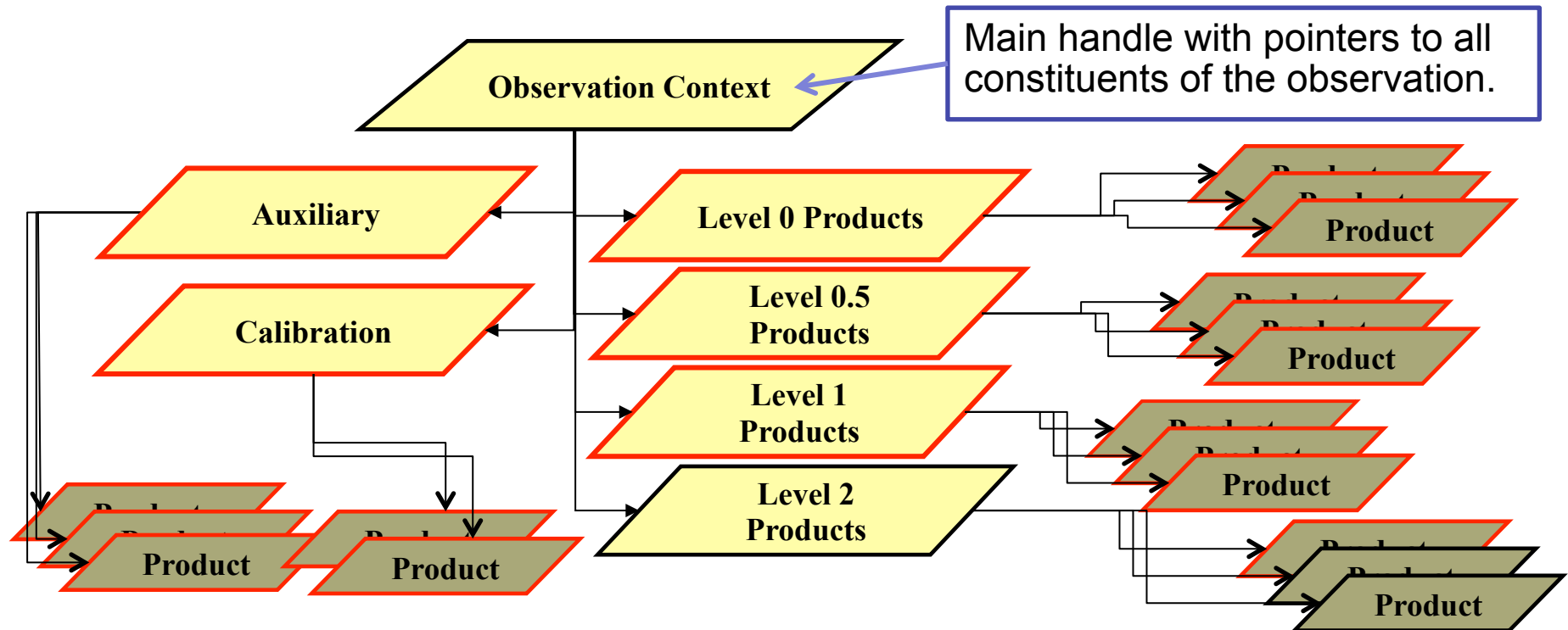
- Contexts are products that point to other products.
- Contexts provide structure to the data (like directories in a file system)



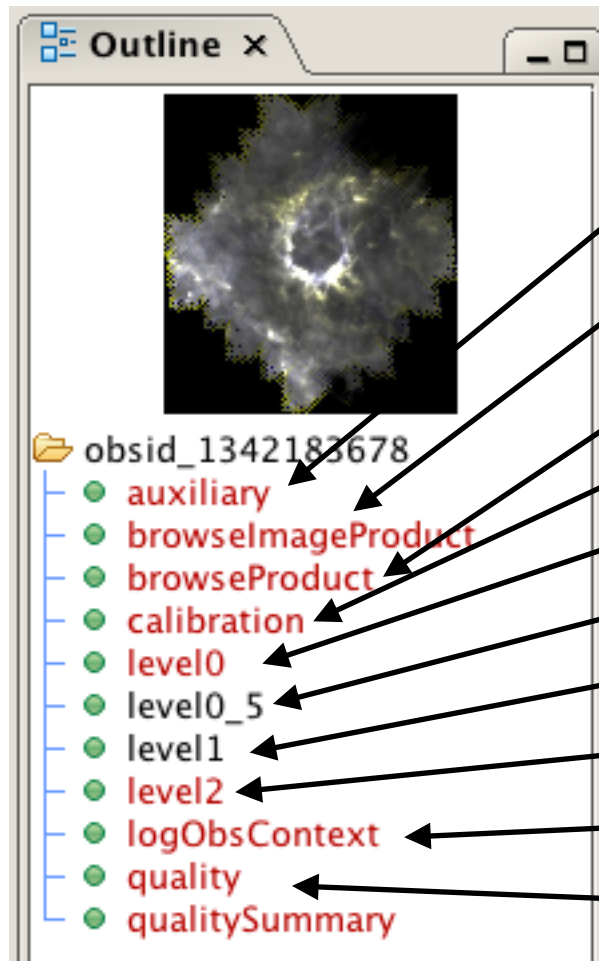


The Observation Context

- Observations are organized in product trees consisting of contexts and products with datasets.
- When loading an observation, initially only the observation context is loaded. Other products/contexts are loaded automatically if necessary (“lazy loading”).
- Products not loaded yet into RAM are shown in red in the outline viewer of HIPE.
- When saving an observation context to a pool, all dependent products are saved as well.



Example: SPIRE Photometer Observation Context



These are all contexts!

Auxiliary data: Pointing, uplink info, etc...

Browse image product: Simple representation of final map for archive quicklook

Browse product: Another browse product

Calibration data: Products used for processing

Level 0: Raw unprocessed reformatted data

Level 0.5: Data converted to engineering units

Level 1: Calibrated flux timelines [Jy/pixel]

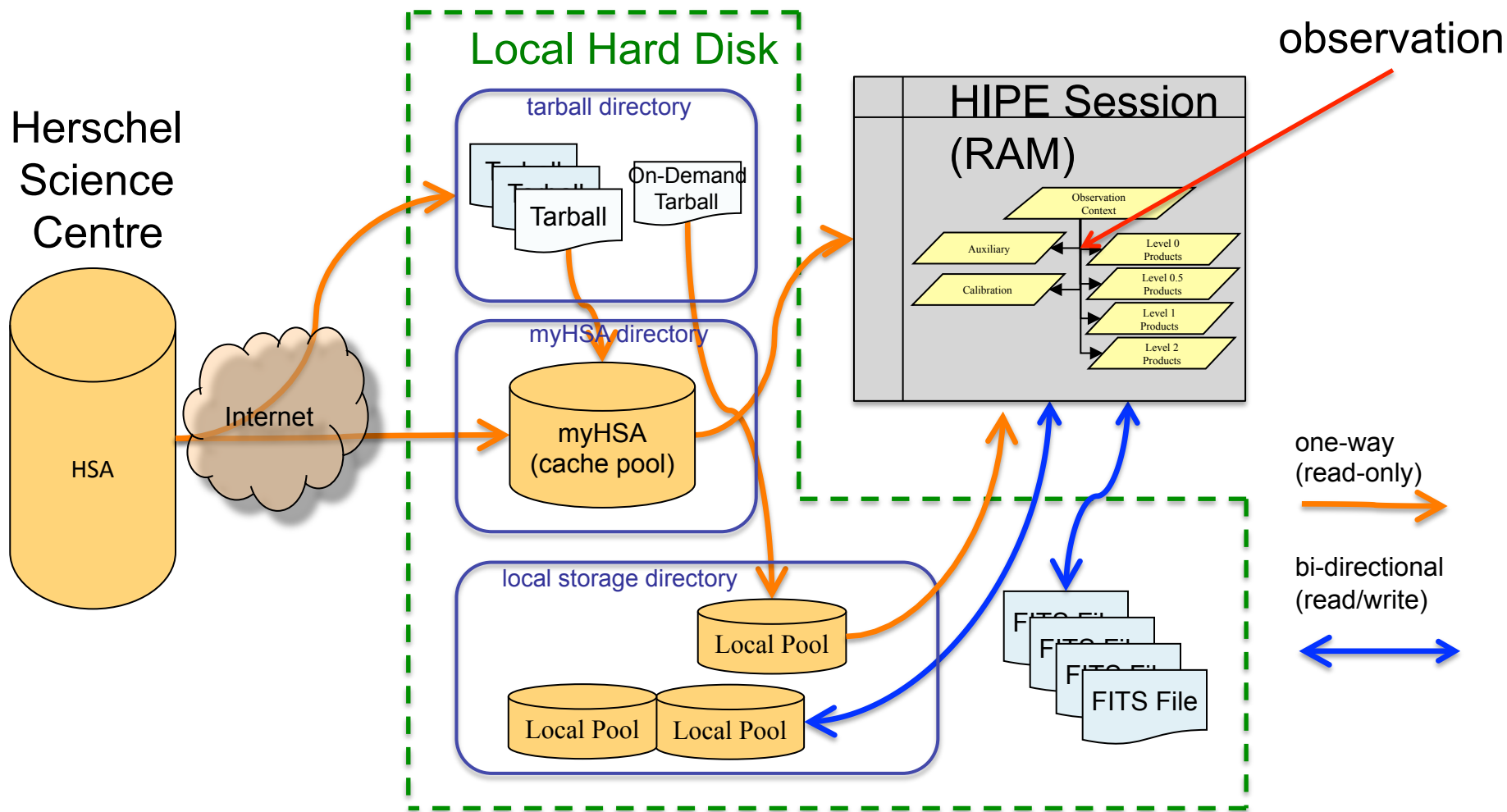
Level 2: Images

Reduction Log: Data reduction history

Quality control data: Tells whether things went O.K.



HIPE Input/Output







Data Input (Gimme my data)

- There are three important methods to load an observation into HIPE.
 - HSA User Interface (HUI)
 - Direct retrieval
 - Tarballs
 - `getObservation()`








Start HSA User
Interface here



Data Input directly from HSA User Interface (HUI)

- Observations can be sent **directly** from the HSA User Interface to the HIPE session.
- Invoke it from the toolbar and select observations. 
- Click on the hard disk symbol and select
 - Send to External Application 
 - All
- The observation context will appear in the variables view of the HIPE session as a **new variable**.

The screenshot shows the HSA Science Archive v4.3.1 interface. An orange arrow points to the 'Send to External Application' option in the context menu. Another orange arrow points to the 'All' option in the sub-menu. The table below shows the data for the selected observation.

Observation ID	Postcards	Target	Duration
1342178903	N/A	NGC7538 IRS1	23h 13
1342179050		M74	01h 36
1342180456		NGC4038/9	12h 01
1342180457		Antennae	12h 01
1342180458		Garrard (C/2008 Q3)	12h 33
1342180459		Garrard (C/2008 Q3)	12h 33
1342180460		Garrard (C/2008 Q3)	12h 33
1342180461		Garrard (C/2008 Q3)	12h 33



Data Input via Tarball

- Observations can be packaged into tarballs, retrieved by FTP, and registered with the myHSA pool of HIPE.
- Click on the **shopping basket** symbol of the observation.
- Go to shopping basket, select observation, and submit request. **put into basket**
- Upon receipt of e-mail retrieve tarball with browser or FTP.
- Expand and place in directory where dataset can stay.
- Find xml file (Saturn Icon) of observation with Navigator and double-click to import.
- Observation appears as variable in the HIPE session and can also be found in the myHSA pool.
- Note: This import step needs only to be performed once. Next time the observation is immediately available using `getObservation()` or Product Browser.

go to shopping basket

The screenshot shows the HSA Science Archive v4.3.1 interface. At the top, there is a navigation bar with 'File', 'View', 'Tools', 'Account', and 'Help'. Below this is a search bar and a shopping basket icon. The main content area displays a table of observations with columns for 'Observation ID', 'Postcards', 'Target', and 'RA/DEC'. An orange arrow points to the shopping basket icon in the top navigation bar, and another orange arrow points to the shopping basket icon in the first column of the observation table. A 'Shopping Basket Overview' window is open, showing a list of observations with one selected: '1342179050 M74'. The status bar at the bottom indicates 'Log Console' and 'beschulz has logged in at 11:12:58 PM'.

Observation ID	Postcards	Target	RA/DEC
1342178903	N/A	NGC7538 IRS1	23h 13m 41.21s +6
1342179050		M74	01h 36m 42.41s +1
1342180456		NGC4038/9	
1342180457		Antennae	
1342180458	N/A	Garradd (C/200	
1342180459		Garradd (C/200	
1342180460		Garradd (C/200	



Data Input via Tarball

- Observations can be packaged into tarballs, retrieved by FTP, and registered with the myHSA pool of HIPE.
- Click on the shopping basket symbol of the observation.
- Go to shopping basket, select observation, and **submit request**.
- Upon receipt of e-mail retrieve tarball with browser or FTP.
- Expand and place in directory where dataset can stay.
- Find xml file (Saturn Icon) of observation with Navigator and double-click to import.
- Observation appears as variable in the HIPE session and can also be found in the myHSA pool.
- Note: This import step needs only to be performed once. Next time the observation is immediately available using `getObservation()` or Product Browser.

HSA Science Archive v4.3.1

File View Windows Account Tools Help

HERSCHEL @esa

Search Observations #1 Shopping Basket

Observations [1 Observation]

Retrieve On Demand Reprocessing

	Observation	Target Name	RA	DEC
<input checked="" type="checkbox"/>	1342179050	M74	01h 36m 42.41s	+15d 46' 34.18"

select

shopping basket

submit request

Submit Request Remove All Close

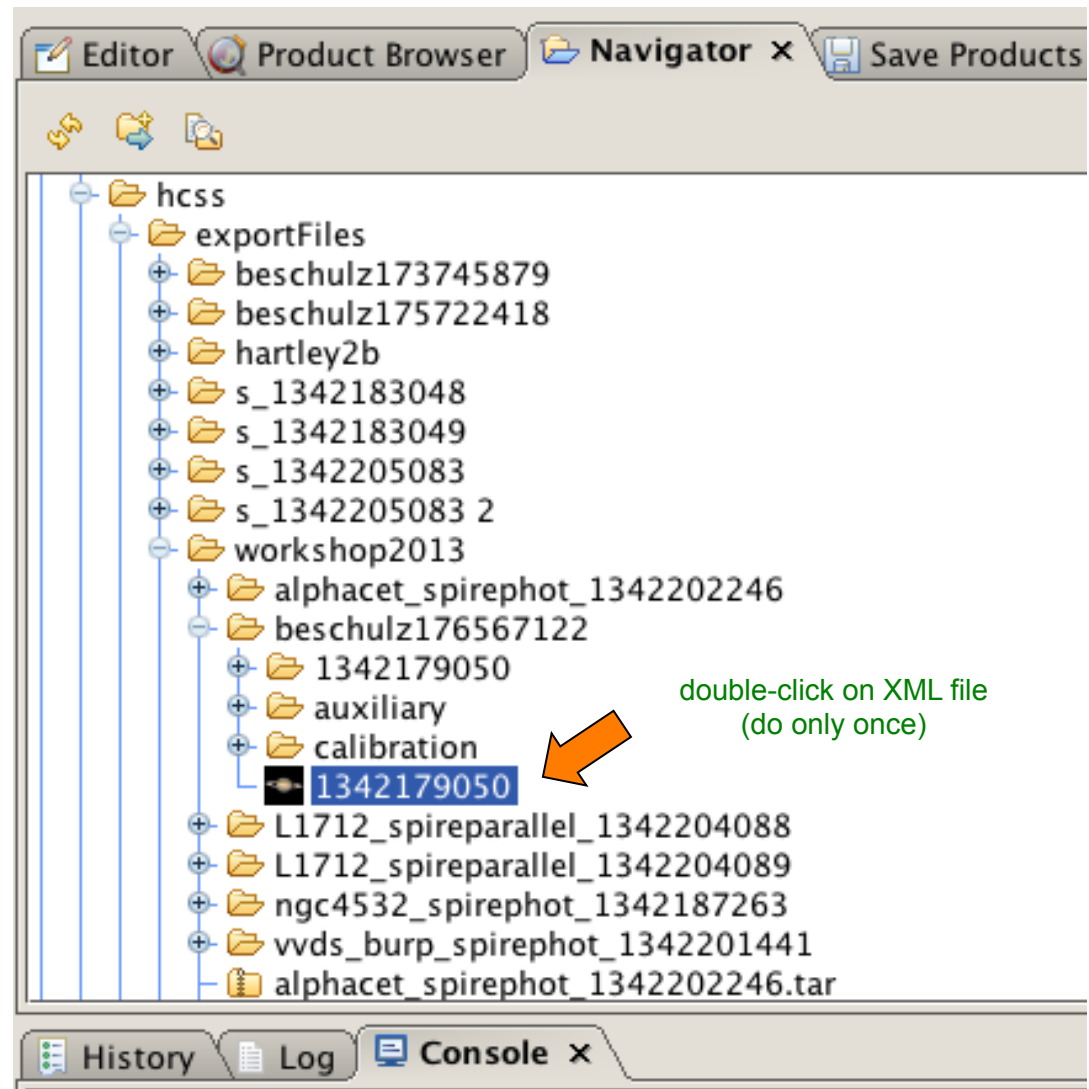
Log Console

beschulz has logged in at 11:12:58 PM



Data Input via Tarball

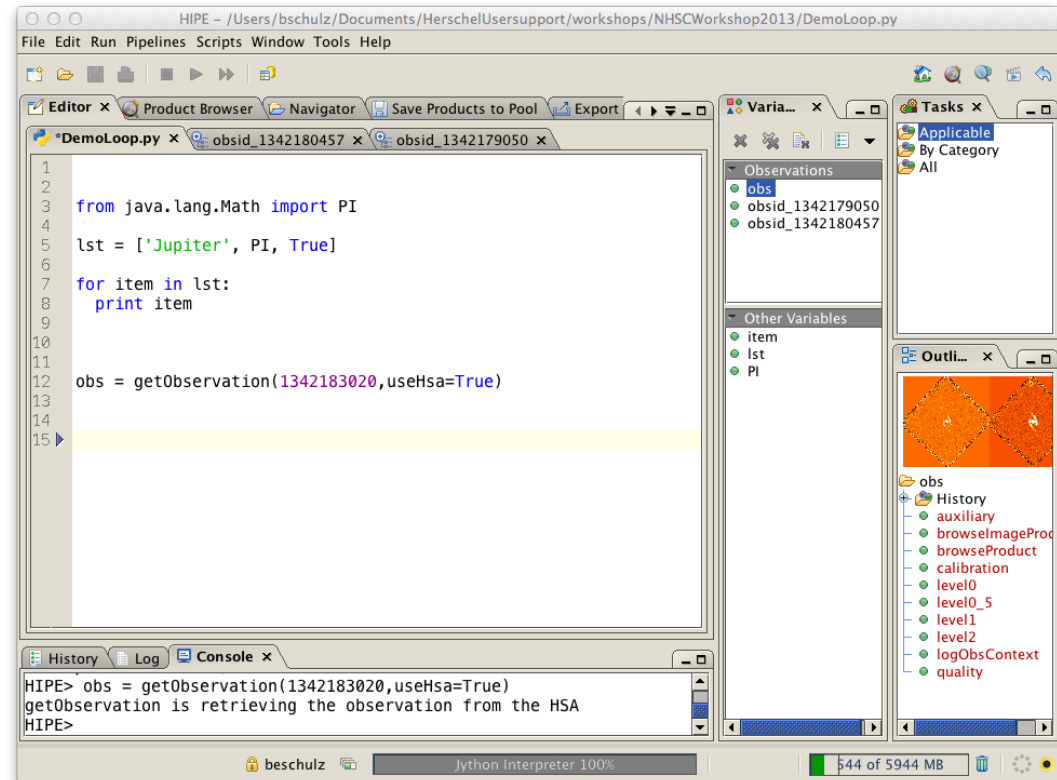
- Observations can be packaged into tarballs, retrieved by FTP, and registered with the myHSA pool of HIPE.
- Click on the shopping basket symbol of the observation.
- Go to shopping basket, select observation, and submit request.
- Upon receipt of e-mail retrieve tarball with browser or FTP.
- Place in directory where dataset can stay and expand it there.
- Find xml file (Saturn Icon) of observation with Navigator and double-click to import.
- Observation appears as variable in the HIPE session and can also be found in the myHSA pool.
- Note: This import step needs **only to be performed once**. Next time the observation is immediately available using getObservation() or Product Browser.





Data Input via getObservation()

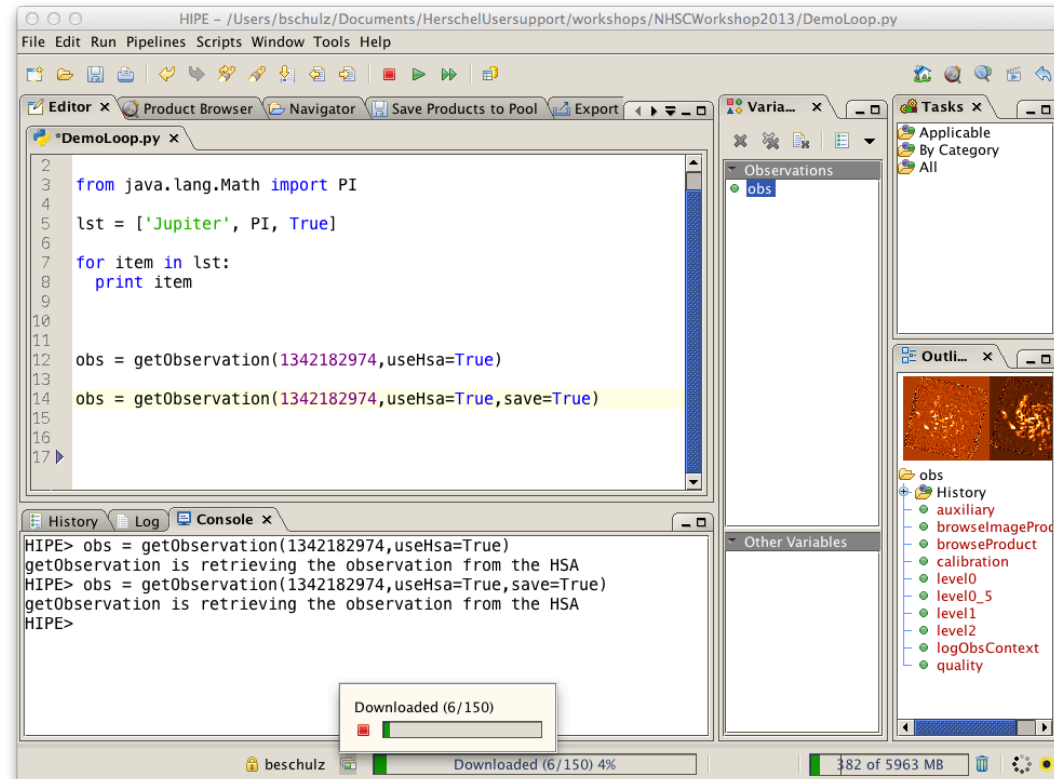
- Observations can be retrieved directly from a script using the getObservation() task.
- With the option useHsa=True, the observation context is retrieved like with “send-to” from the HSA interface and placed into the HIPE session.
- With the additional option save=True, the entire observation is retrieved and saved in the myHsa pool. Next time it will already be on disk.
- This operation happens in the background so you can keep working.





Data Input via getObservation()

- Observations can be retrieved directly from a script using the `getObservation()` task.
- With the option `useHsa=True`, the observation context is retrieved like with “send-to” from the HSA interface and placed into the HIPE session.
- With the additional option `save=True`, the entire observation is retrieved and saved in the `myHsa pool`. Next time it will already be on disk.
- This operation happens in the background so you can keep working.

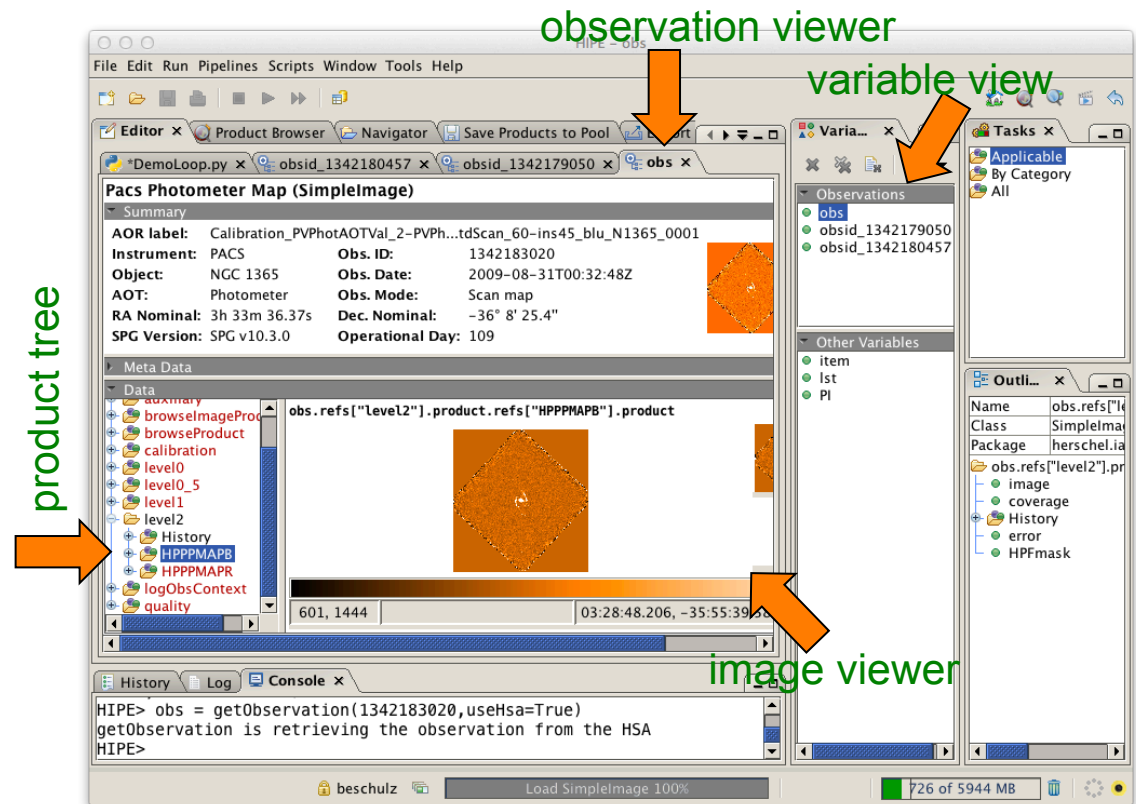




Inspecting Products

- Double-click on the observation context in the Variables View opens the Observation Viewer.
- Dependent contexts and products can be inspected by selection in the product tree to the left.
- Appropriate viewers for datasets are started automatically in the panel to the right of the product tree.
- Note that products only get actually transferred from the HSA when they are looked at (lazy-loading).

Observation Viewer (PACS example)





Data Viewing and Further Proceeds

- More viewers are available and are often specific to a given instrument.
- These will be treated in the respective instrument sessions.
- Sometimes the data inspection reveals shortcomings in the data processing and the astronomer wishes to perform a **data re-processing** in HIPE.
- Data re-processing is usually done in Jython pipeline scripts or by specific “Tasks”.



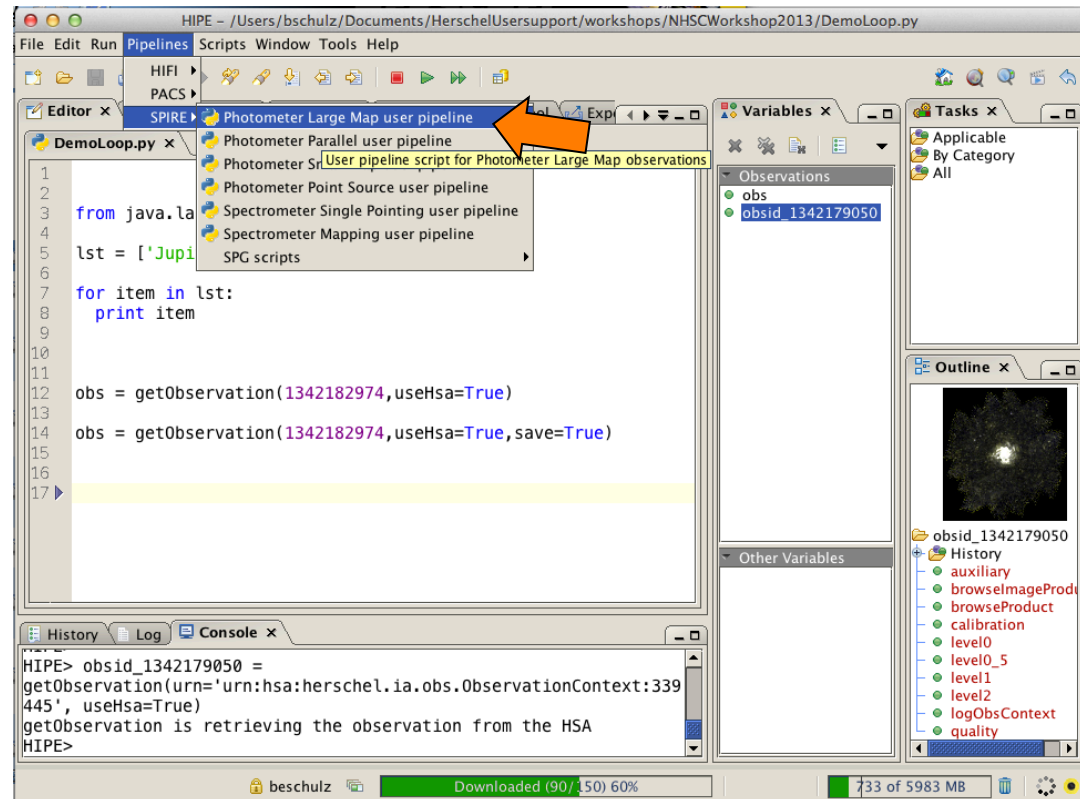
Running a Pipeline Script

- Some custom scripts are available in the HIPE menu already.
- Locations:
 - Under “**Pipelines**” the main pipeline scripts can be found that are used to produce the archive results.
 - Under “**Scripts**” more Jython scripts are available for more specialized non-standard purposes.
- Script Execution in the Editor View:
 - The **single green arrow** executes one statement or block at the lowest level.
 - The **green double arrow** executes the entire file.
 - Indented blocks like for-loops or if-then statements can not be stepped through line by line.
 - The **pause() statement** sets break-points and helps with debugging inside of indented blocks.
 - Single lines or groups of lines inside blocks can be executed with limitations by marking them blue and hitting the single green arrow.



Example: Running the SPIRE Pipeline

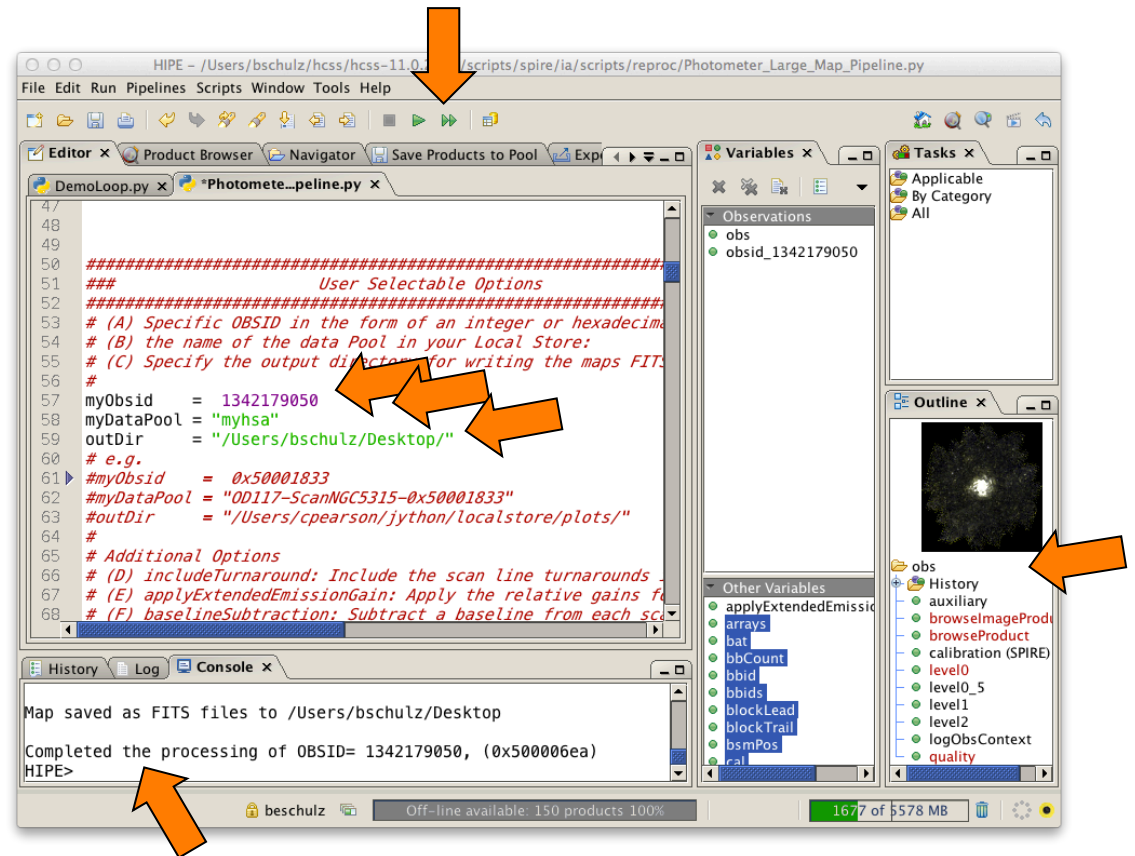
- Select in the Pipelines Menu: “Photometer Large Map user pipeline”
- Edit the script without saving and insert
 - myObsid=1342182974
 - myDataPool='myhsa'
 - outDir='<output directory>'
- Run the script by hitting the green double arrow in the top toolbar.
- After a while the updated observation will appear as obs under Variables and in the Outline view.





Example: Running the SPIRE Pipeline

- Select in the Pipelines Menu: “Photometer Large Map user pipeline”
- Edit the script without saving and insert
 - myObsid=1342182974
 - myDataPool='myhsa'
 - outDir='<output directory>'
- Run the script by hitting the green double arrow in the top toolbar.
- After a while the updated observation will appear as obs under Variables and in the Outline view.





Saving Products in FITS Files

- If only a few products are of interest after processing is complete, each product can be saved to a FITS file.
 - Right-click on the product (e.g. a map) and select Send-to → FITS-file.
 - In the upcoming GUI of the task simpleFitsWriter() enter a filename with path and hit the “Accept” button.
 - The equivalent command appears in the Command View and can be re-used in a Jython script.



Saving Products to FITS files

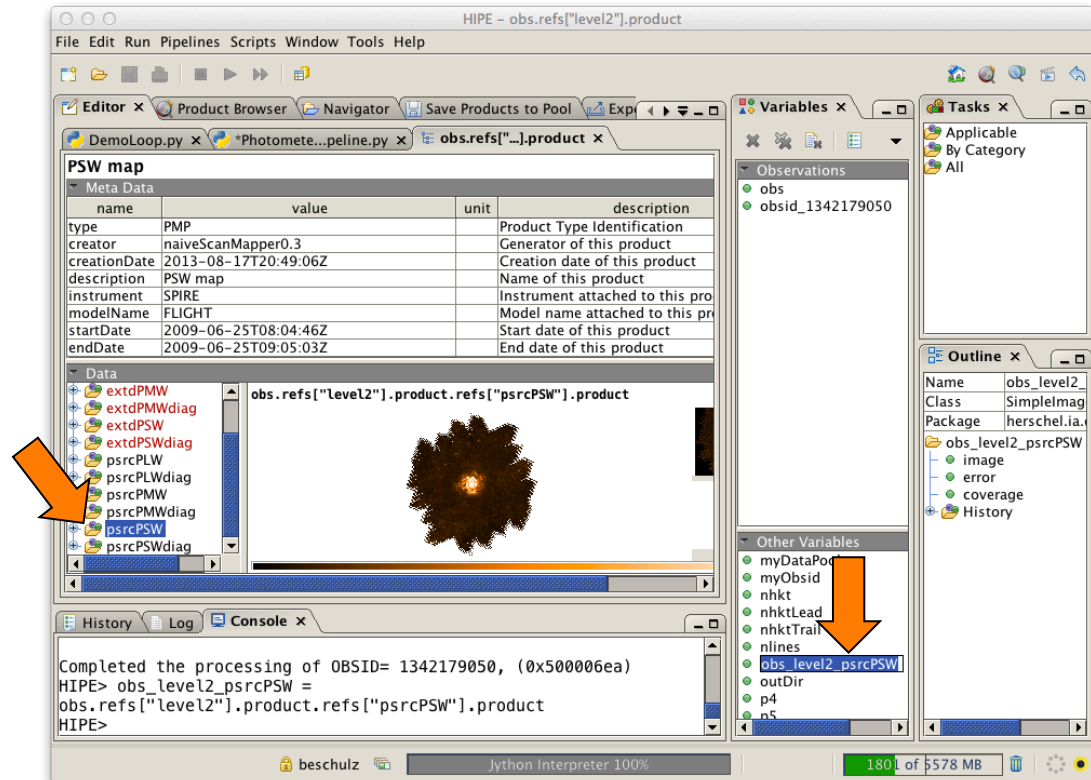
- The script already saved the three map products as FITS files.
- To try using the GUI:
- Double-click Level 2 in the Outline view.
- Drag a map from Level 2 into the variable space.
- Right click on the new variable and select “Send-to” and “FITS-file”.
- In the upcoming GUI enter a filename with path and hit the “Accept” button.
- Note that you just used a Task from its default GUI (more about this later).
- Note also that the scripted version of what you just did appears in the Console View.

```
HIPE - /Users/bschulz/hcss/hcss-11.0.2934/scripts/spire/ia/scripts/reproc/Photometer_Large_Map_Pipeline.py
File Edit Run Pipelines Scripts Window Tools Help
Editor x Product Browser Navigator Save Products to Pool Export Herschel data from HIPE Calibrators
DemoLoop.py x *Photomete...eline.py x
373 if obs.level2.refs['PLW']!=None: obs.level2.refs.remove('PLW')
374 obs.level2.setProduct("psrcPSW", mapPsw)
375 obs.level2.setProduct("psrcPMW", mapPmw)
376 obs.level2.setProduct("psrcPLW", mapPlw)
377 #
378 print "Finished the map making for OBSID= %i, (0x%x)"%(myObsid,myObsid)
379
380 print
381 #
382 # -----
383 # Save Maps to output directory
384 simpleFitsWriter(mapPsw, "%smapPSW_%i.fits"%(outdir, myObsid))
385 simpleFitsWriter(mapPmw, "%smapPMW_%i.fits"%(outdir, myObsid))
386 simpleFitsWriter(mapPlw, "%smapPLW_%i.fits"%(outdir, myObsid))
387 print "Map saved as FITS files to %s"%(outdir)
388 #
389 ### Finished the Mapmaking ###
390 #####
391 # Finally we can save the new reprocessed observation back to your hard disk
392 # Uncomment the next line and choose a poolName, either the existing one or a new one
393 #
394 #saveObservation(obs,poolName="enter-a-poolname",saveCalTree=True)
395 #
396 #
397 #
398 print
399 print "Completed the processing of OBSID= %i, (0x%x)"%(myObsid,myObsid)
400
```



Saving Products to FITS files

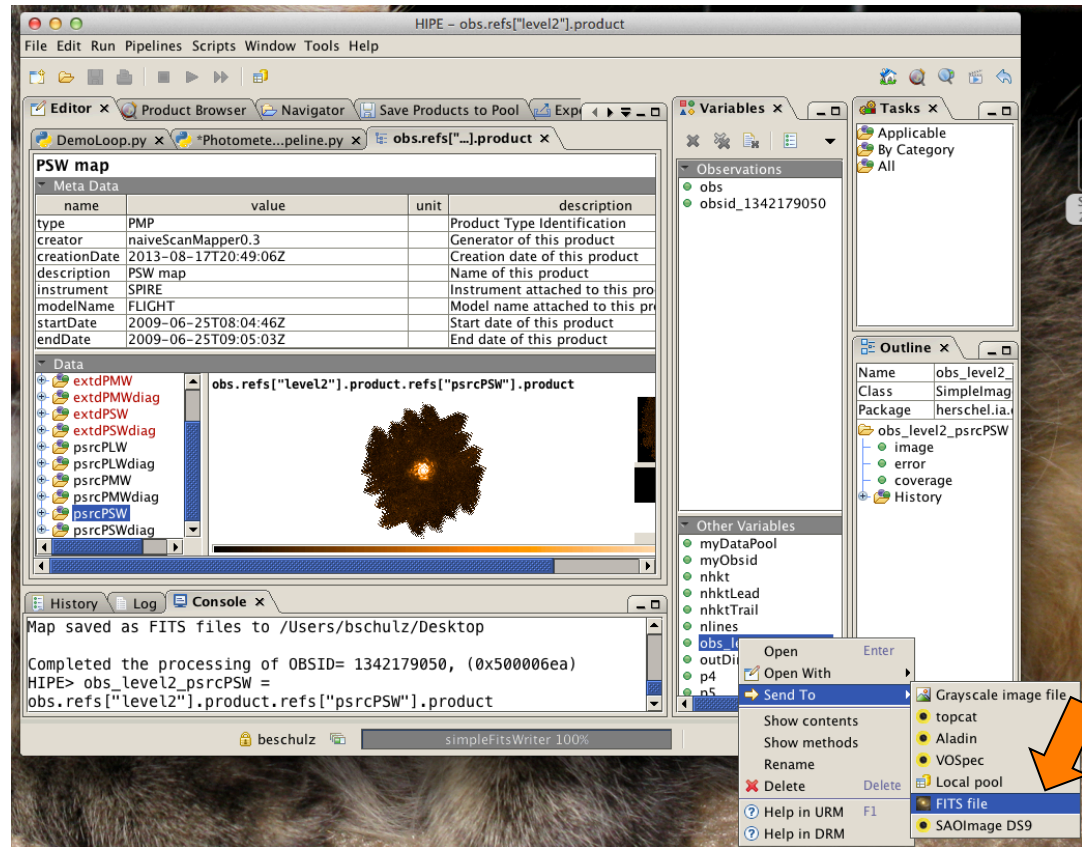
- The script already saved the three map products as FITS files.
- To try using the GUI:
- Double-click Level 2 in the Outline view.
- Drag a map from Level 2 into the variable space.
- Right click on the new variable and select “Send-to” and “FITS-file”.
- In the upcoming GUI enter a filename with path and hit the “Accept” button.
- Note that you just used a Task from its default GUI (more about this later).
- Note also that the scripted version of what you just did appears in the Console View.





Saving Products to FITS files

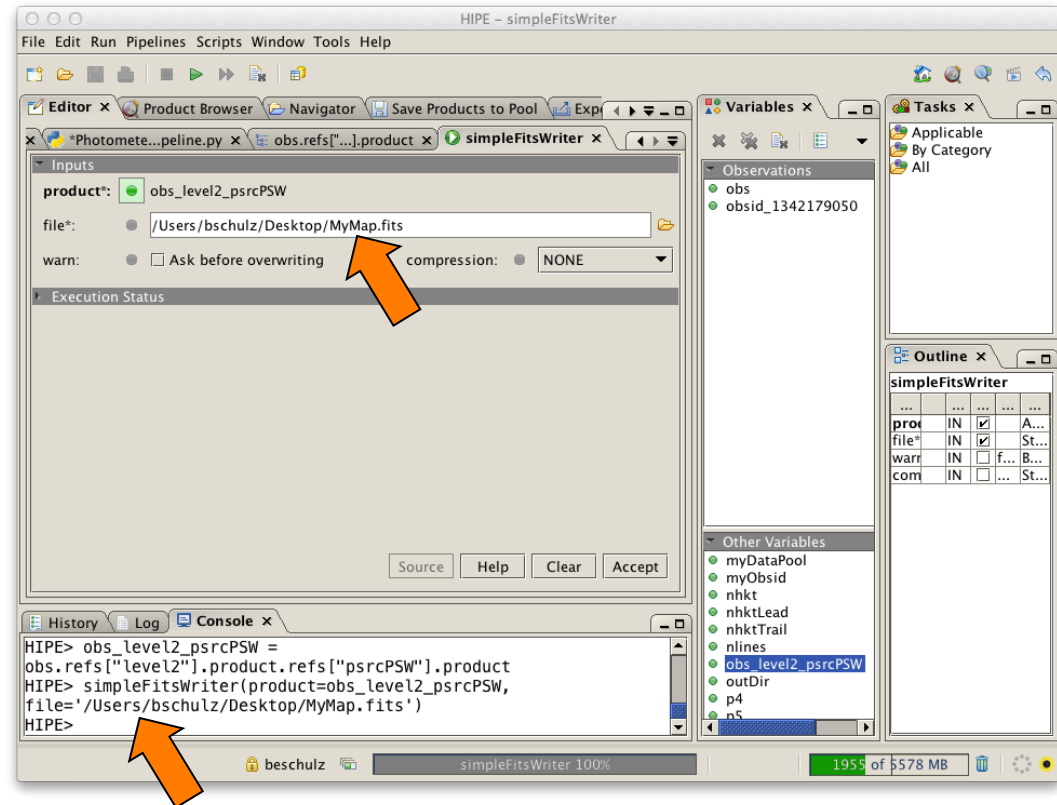
- The script already saved the three map products as FITS files.
- To try using the GUI:
- Double-click Level 2 in the Outline view.
- Drag a map from Level 2 into the variable space.
- Right click on the new variable and select “Send-to” and “FITS-file”.
- In the upcoming GUI enter a filename with path and hit the “Accept” button.
- Note that you just used a Task from its default GUI (more about this later).
- Note also that the scripted version of what you just did appears in the Console View.





Saving Products to FITS files

- The script already saved the three map products as FITS files.
- To try using the GUI:
- Double-click Level 2 in the Outline view.
- Drag a map from Level 2 into the variable space.
- Right click on the new variable and select “Send-to” and “FITS-file”.
- In the upcoming GUI enter a filename with path and hit the “Accept” button.
- Note that you just used a Task from its default GUI (more about this later).
- Note also that the scripted version of what you just did appears in the Console View.





Saving Observations in Local Pools

- If you work on an observation and **update many of its products**, and you want to save a copy of your work for the next session, saving single FITS files is impractical.
- In this case the better way is to use a **Local Pool**.
- Local Pools are **databases for products** and contexts that keep the **product tree structure intact**.
- Saving and retrieving entire observation trees can be performed very easily using the **observation context as a handle**.



Saving Observations in Pools

- Select the observation context in the **Variable View**.
- Select “Send To” and “Local pool”.
- Find observation in list.
- Add some text (Tag) for identification.
- Enter a destination pool name (not existing ones will be created new).
- Hit ‘Save’ button.
- Note the command line representation:

```
saveProduct (product=obs,  
pool='myTestPool',  
tag='obs 1342179050 my  
last reduction')
```

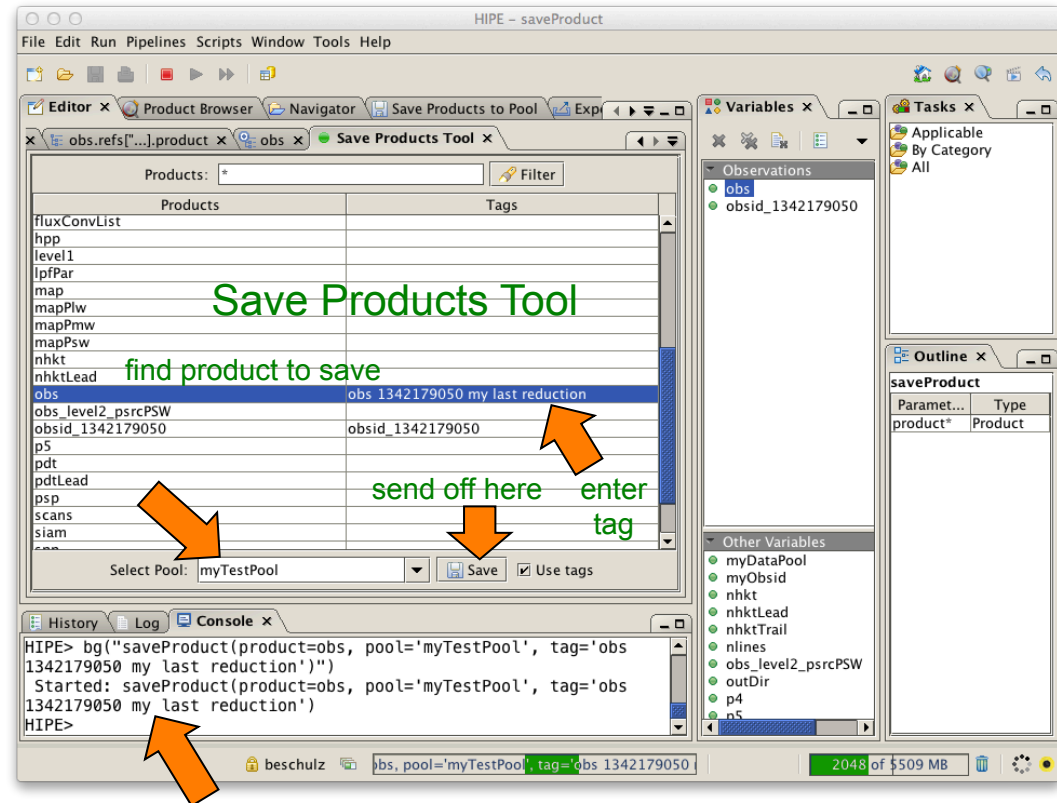
Variables View

The screenshot shows the HIPE software interface. The main window displays the 'ObservationContext for SPIRE data of observation 1342179050'. The 'Variables View' is open on the right, showing a list of variables. The 'Send To' menu is open, and the 'Local pool' option is selected. The console window at the bottom shows the command line representation of the saveProduct function.



Saving Observations in Pools

- Select the observation context in the Variable View.
- Select “Send To” and “Local pool”.
- Find observation in list.
- Add some text (Tag) for identification.
- Enter a destination pool name (not existing ones will be created new).
- Hit ‘Save’ button.
- Note the command line representation:
`saveProduct (product=obs,
pool='myTestPool',
tag='obs 1342179050 my
last reduction')`



command line form uses
task saveProduct()



The Product Browser

- The Product Browser can be found in “Window” as a view under “Data Access”.
- The main panels are “Data Source”, “Search parameters”, and “Query Result Display”
- Data Source shows all the pools in your storage directory and the myHsa pool.
- On the right some search criteria can be chosen.
- The “On-line” button enables querying the HSA through the internet if an observation isn’t present locally in the MyHSA pool.

On-line
Off-line
selector

Search Parameters

On-line
Off-line
selector

Data Source (Pools)

Query Result Display

Pool	obsid	odNumber	tag	object	total size	aot	obsMode
125 results found							

History Log Console x

HIPE>
HIPE>
HIPE>
HIPE>
HIPE>

beschulz Jython Interpreter 100%



Managing Observations and Pools

- Pools can be created, inspected, renamed, moved, exported, and deleted.
- The **Product Browser** helps to easily visualize and manage the content of pools.
- The myHSA pool is special:
 - MyHsa acts as a **local cache** between the HSA at the Herschel Science Center and your HIPE session on your computer.
 - All pools can be written to, except for the “myHsa” pool.
 - The other pools are intended for saving results that are different from those in the HSA, i.e. products from **your own processing and analysis**.
 - The data transfer between the HSA and the myHSA pool is optimized.
 - Saving observations that were retrieved straight from the HSA into a pool, is less efficient.
- The Product Browser can be used to retrieve observations from the archive by querying myHsa with the On-line option on.



Pool Content and Retrieval

- Find and select the newly created pool “myTestPool”.
- Hit the “Run” button.
- The query result list will show one line representing the observation that was just saved.
- Double-clicking the observation will retrieve the observation context.
- Note the more complex script equivalent in the Console View.

The screenshot shows the HIPE software interface with several panels. The 'Product Browser' panel on the left lists various pools, with 'myTestPool' selected. The 'Run' button is highlighted with an orange arrow. The 'Query Result' table below shows one result for 'myTestPool' with observation ID 1342179050. The 'Console' panel at the bottom displays a complex script for querying the pool. The 'Observations' panel on the right shows the selected observation and its context, with an orange arrow pointing to the observation ID. The 'Other Variables' panel on the right lists various variables available for the observation.

Pool	obsid	odNumber	tag	object	total size	aot
myTestPool	1342179050	42	obs 1342179050 my l...	M74	935942440	Photometer

```
HIPE> QUERY_RESULT =  
ProductStorage([PoolManager.getPool('myTestPool')]).select(herschel.ia.p  
al.query.MetaQuery(herschel.ia.obs.ObservationContext, "p", "1"))  
HIPE> # Added variable: QUERY_RESULT  
HIPE> # Added variable: QUERY_RESULT  
HIPE> # Added variable: selected  
HIPE> obsid_1342179050_1 = QUERY_RESULT[0].product  
HIPE>
```



MyHSA Access

- To access observations in the **local** myHSA pool, select MyHSA
- Ensure “Off-line” is selected.
- Hit the “Run” button.
- Note the presence of the earlier imported observations from SPIRE, HIFI and PACS.

The screenshot shows the HIPE software interface with the following components:

- Product Browser:** Shows a list of data sources: MyHSA, DestriperL2Degl, and DestriperTest. The 'MyHSA' source is selected. Below it, the 'Off-line' radio button is selected.
- Run Button:** A button labeled 'Run' is visible below the Product Browser.
- Query Result Table:** A table with 125 results found. The columns are: Pool, obsid, odNumber, tag, object, total size, aot, obsMode, and instrument. The table lists observations from the 'hsa' pool, including objects like M 101, NGC 1365, Antennae, M74, and M83, with instruments like PACS, HIFI, and SPIRE.
- Console:** Shows the execution of a query: `HIPE> obsid_1342179050_1 = QUERY_RESULT[0].product` and `HIPE> QUERY_RESULT1 = ProductStorage([PoolManager.getPool('myhsa')]).select(herschel.ia.pal.query.MetaQuery(herschel.ia.obs.ObservationContext, "p", "1"))`.



On-line Archive Access Example

- Let's find all parallel mode PACS observations on observational days 600 and 605.
- Switch to "On-line"
- Enter "PACS" into Instrument.
- Use the + sign to add two more "Observational Day" search criteria and one "Observation Mode".
- Select appropriate logical operators and numbers.
- Enter "Parallel Mode" into Observation Mode
- Hit "Run".
- Select "all" to show the full list of results.

The screenshot shows the HIPE software interface with the following search parameters:

- On-line (selected)
- Instrument: PACS
- Operational Day (odNumber) >= 600
- Operational Day (odNumber) <= 605
- Observation Mode (obsMode) == Parallel Mode

The results table shows 9 results found:

Pool	obsid	odNumber	tag	object	total size	aot	obsMode	instrume
hsa	1342212300	600		DC300-17	5222699...	Parallel M...	Parallel M...	PACS
hsa	1342212302	600		V1	1551557...	Parallel M...	Parallel M...	PACS
hsa	1342212408	605		V4	1607408...	Parallel M...	Parallel M...	PACS
hsa	1342212301	600		V1	1613639...	Parallel M...	Parallel M...	PACS
hsa	1342212407	605		V4	1546483...	Parallel M...	Parallel M...	PACS
hsa	1342212378	604		V3	1549685...	Parallel M...	Parallel M...	PACS
hsa	1342212358	603		V3	1609189...	Parallel M...	Parallel M...	PACS
hsa	1342212379	604		V3	1610537...	Parallel M...	Parallel M...	PACS

The console window shows the following output:

```

HIPE> # Added variable: QUERY_RESULT6
HIPE> # Added variable: QUERY_RESULT6
HIPE> # Added variable: QUERY_RESULT6
HIPE> # Added variable: QUERY_RESULT6
HIPE>
    
```



Managing Observations

- Find all observations in all local pools.
- De-select “MyHSA” by clicking on another pool.
- Select “Local Pools”.
- Hit “Run”.
- Right click on an observation.
- The menu allows to:
 1. load an observation context into RAM (create variable),
 2. Delete an observation from a pool (Remove product....),
 3. Export FITS (not useful because it is just a context.).

The screenshot shows the HIPE software interface. The 'Product Browser' window is active, displaying a list of observations. The 'Local Pools' section is selected, and the 'Run' button is visible. A context menu is open over a selected observation, showing options like 'Create variable', 'Remove product from storage/pool', and 'Export FITS...'. The table below shows the following data:

Pool	obsid	odNumber	tag	object	total size	aot	obsM
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	30891804	Photometer	Small
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	29474053	Photometer	Small
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	30958355	Photometer	Small
parallelTestMaps	1342251957	1236	1342251...	Pipe_fillg...	545254467	Parallel M...	Paralle
parallelTestMaps	1342251956	1236	1342251...	Pipe_fillg...	538522324	Parallel M...	Paralle
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	30877008	Photometer	Small
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	29466469	Photometer	Small
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	30874388	Photometer	Small

The console window at the bottom shows the following output:

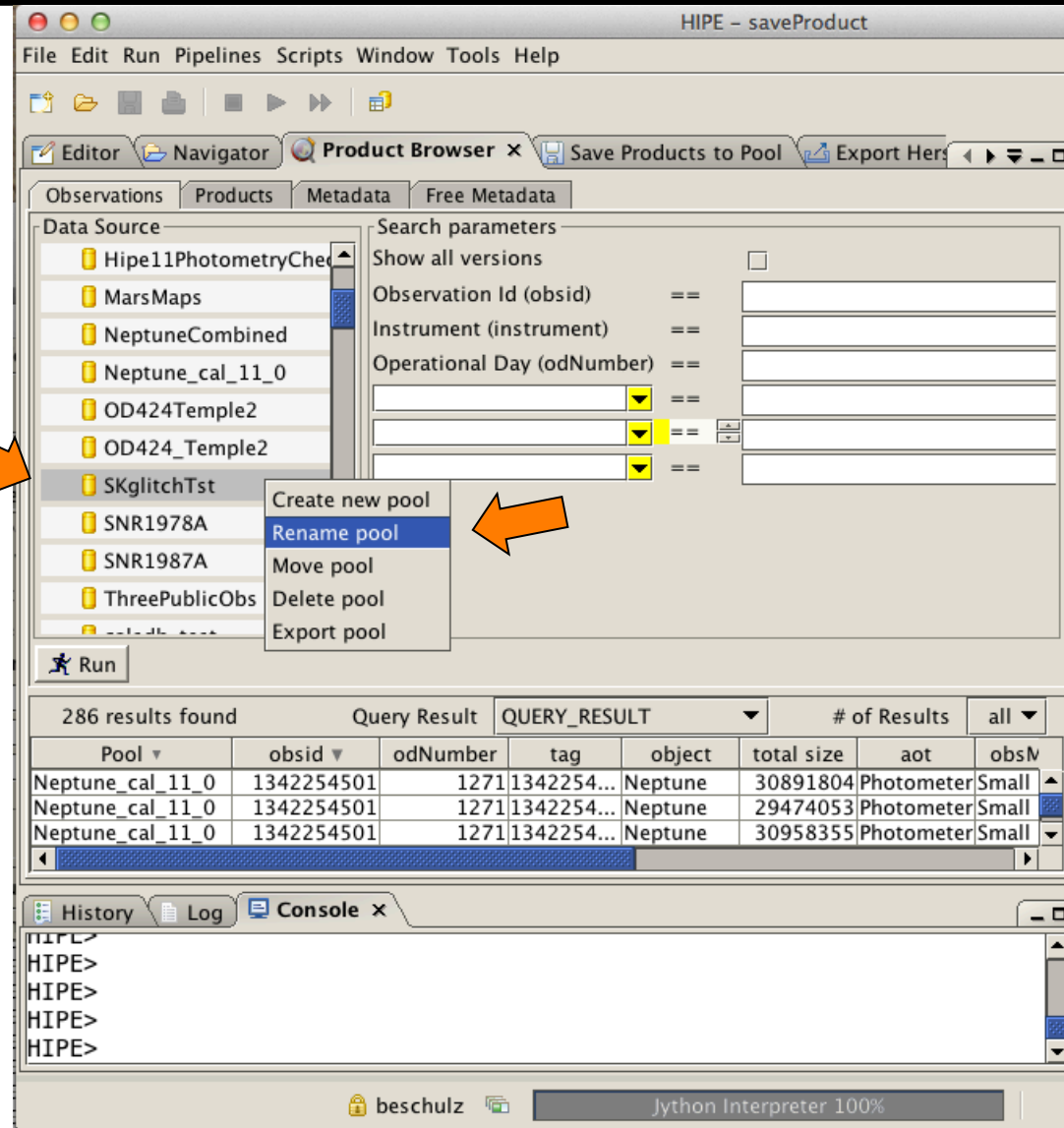
```

HIPE> # Added variable: QUERY_RESULT
HIPE> # Added variable: QUERY_RESULT
HIPE> # Added variable: selected
HIPE> # Added variable: selected
HIPE>
    
```



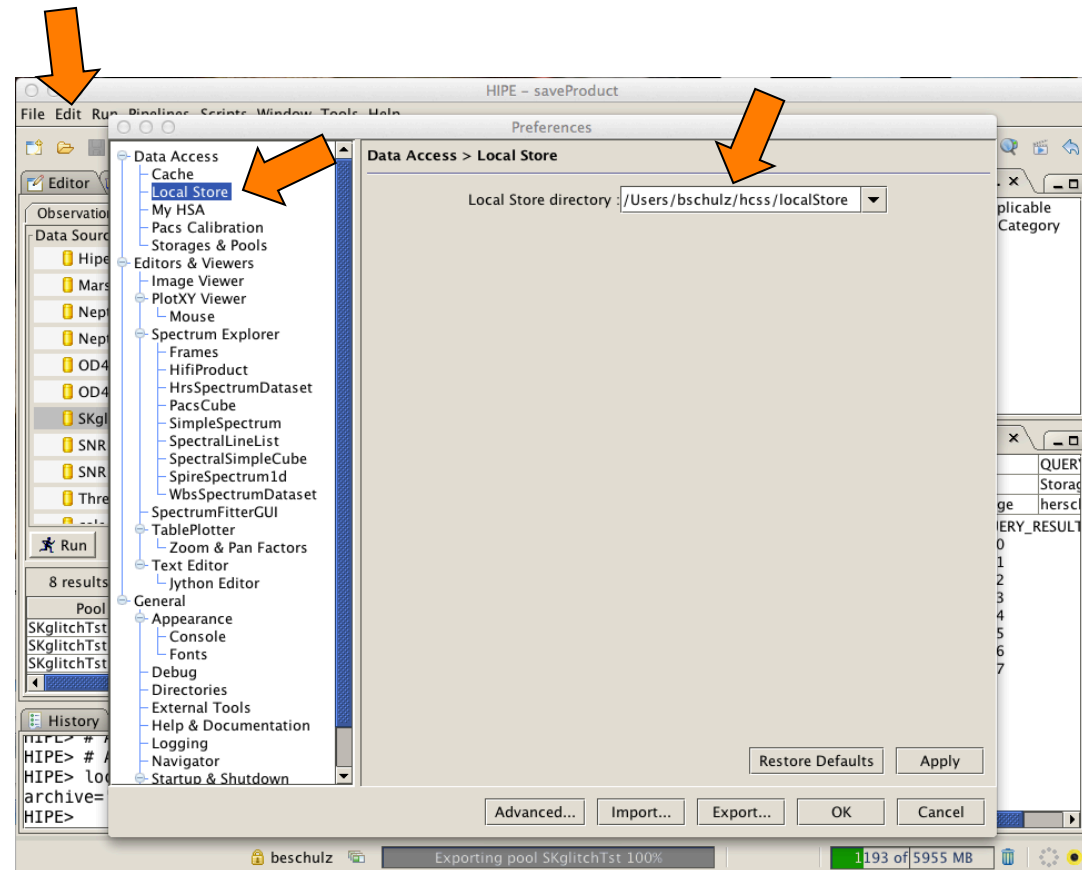

Managing Pools

- Select a pool and right-click.
- Pools can be **created, renamed, moved to another storage directory, deleted, and exported.**
- Moving will move the pool to another directory in your file system.
- Export will copy the contents of a pool into a zip file, so it can be sent to collaborators and other pundits.



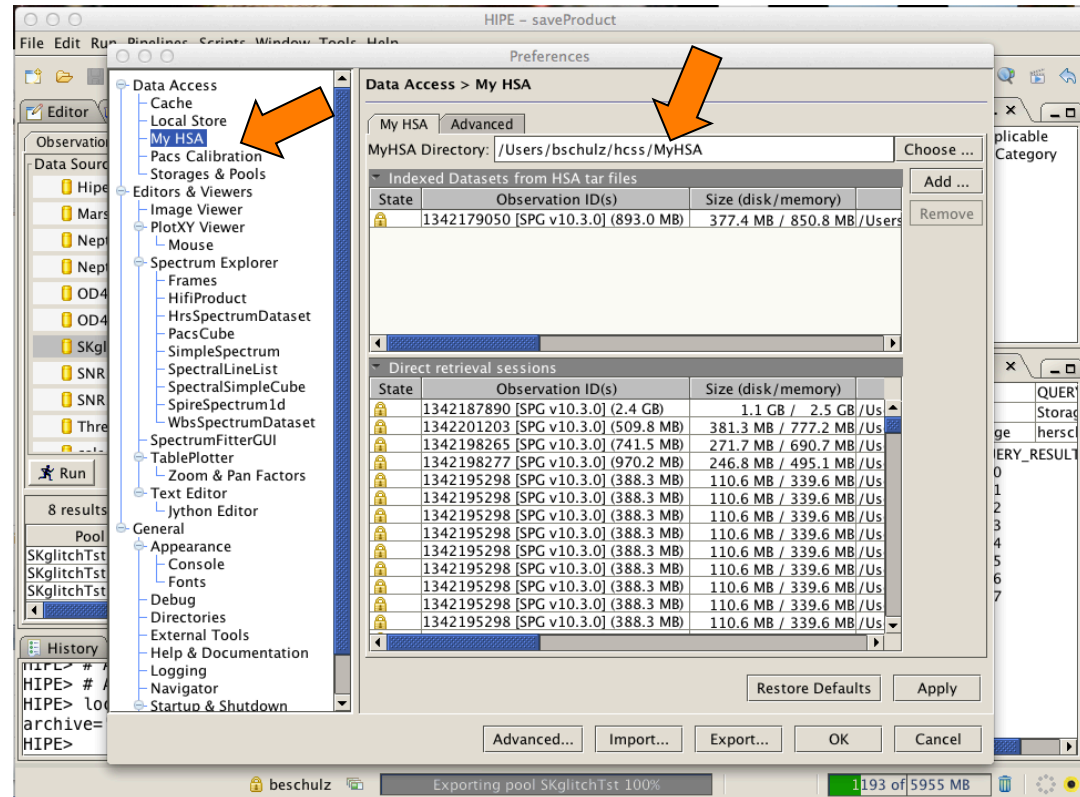
A Place for Ordinary Pools

- The location of the directory where the pools are stored can be determined in the “Preferences” menu.
- Select “Edit” → “Preferences”.
- Select there “Data Access” → “Local Store”.
- It makes sense to configure this to be on a large disk with sufficient space.



A Place for the MyHsa Pool

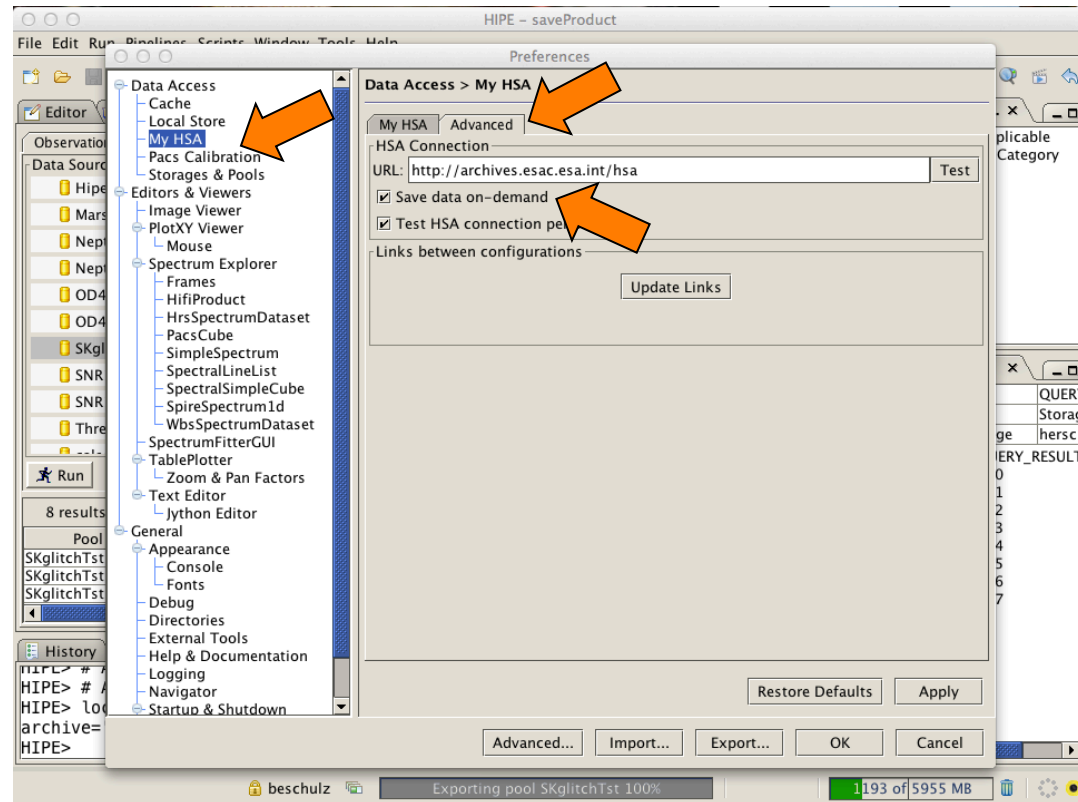
- The **myHsa pool** has a separate directory that can be configured also in “Preferences”
- Select “Edit” → “Preferences”.
- Select there “Data Access” → “My HSA”.
- It is good to choose a disk with sufficient space, when the “**save**” option in `getObservation()` is used or the **save data on-demand** option is activated in the “Advanced” tab.





On-Demand Save Option

- The **save data on-demand option** keeps copies of all products that are loaded through “lazy-loading” in the myHsa pool, making access to the same products **faster next time**.

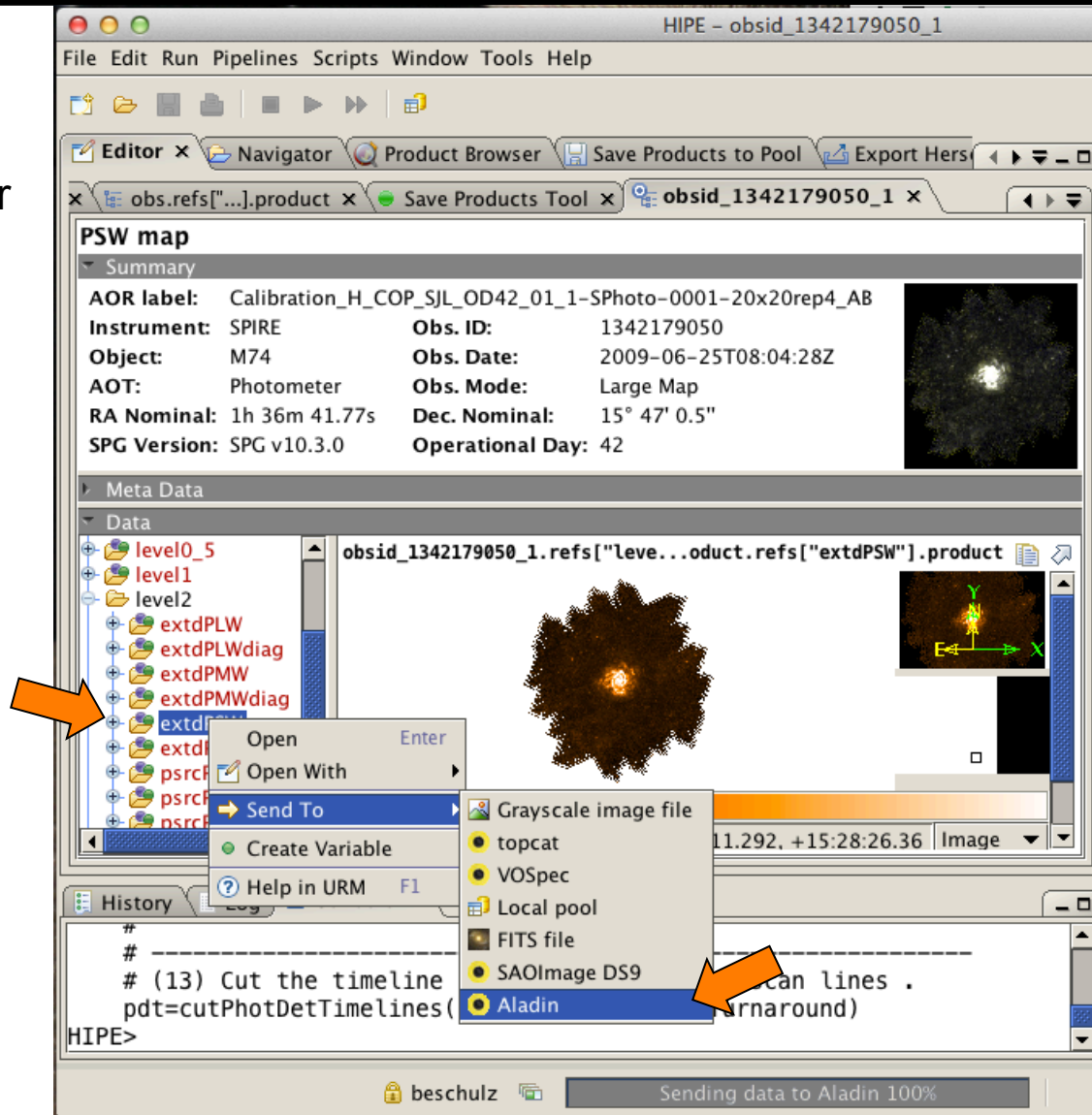


Save data on-demand is off by default



Data Exchange via SAMP (Images)

- HIPE provides support for a powerful interface to share data between applications, the **Simple Application Messaging Protocol (SAMP)**
- The send-to menu provides options to send image data or tabular data to **Topcat, VOSpec, DS9, or Aladin.**
- Right click on a map and send it to Aladin.
- The Java application is downloaded and opens the image.
- Many more visualization and analysis methods become available.

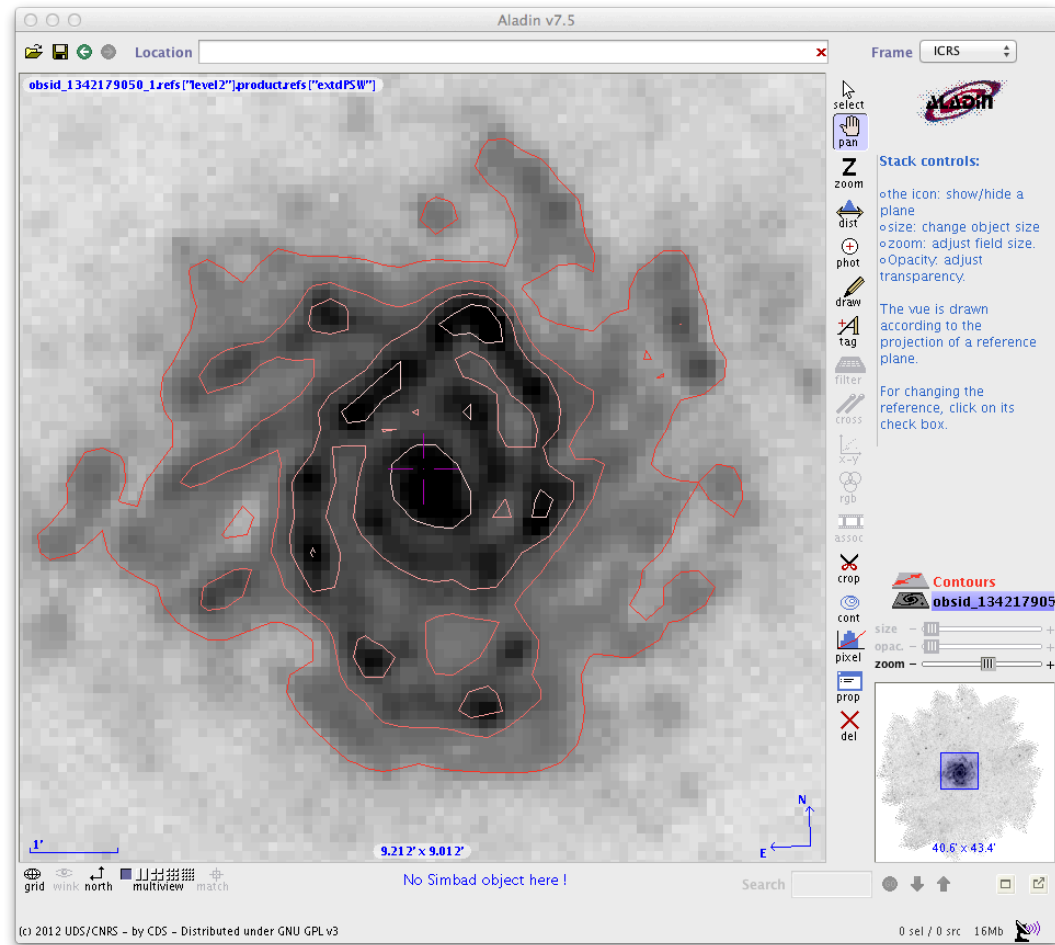


See: <http://www.ivoa.net/documents/SAMP/>



Data Exchange via SAMP (Images)

- HIPE provides support for a powerful interface to share data between applications, the Simple Application Messaging Protocol (SAMP)
- The send-to menu provides options to send image data or tabular data to Topcat, VOSpec, DS9, or Aladin.
- Right click on a map and send it to Aladin.
- The Java application is downloaded and opens the image.
- **Many more** visualization and analysis methods become available.





Data Exchange via SAMP (Tables)

- Another example sending a table dataset.
- Start Topcat on your computer.
- Select a SPIRE diagnostic table and send it to Topcat.
- It appears as a table and can be analyzed using the Topcat features.
- It can even be modified on Topcat and sent back to HIPE using the “Transmit table to all applications using SAMP” button.

HIPE – obsid_1342179050_1

File Edit Run Pipelines Scripts Window Tools Help

Editor x Navigator Product Browser Save Products to Pool Export Here

obs.refs["...].product x Save Products Tool x obsid_1342179050_1 x

Diagnostic Tabledataset for Destriper

Summary

AOR label: Calibration_H_COP_SJL_OD42_01_1-SPhoto-0001-20x20rep4_AB
Instrument: SPIRE Obs. ID: 1342179050
Object: M74 Obs. Date: 2009-06-25T08:04:28Z
AOT: Photometer Obs. Mode: Large Map
RA Nominal: 1h 36m 41.77s Dec. Nominal: 15° 47' 0.5"
SPG Version: SPG v10.3.0 Operational Day: 42

Meta Data

Data

Index	channelName	detIndex	scanNumber	iter	ch
0	PMWA1	271	0	8	2.551341
1	PMWA2	268	0	7	1.846141
2	PMWA3	267	0	2	1.959838
3	PMWA4	285	0	4	1.883377
4	PMWA5	278	0	2	1.873610
		275	0	1	1.913792
		274	0	1	2.326866
		272	0	1	2.039991
			0	6	1.742743

History Log Console x

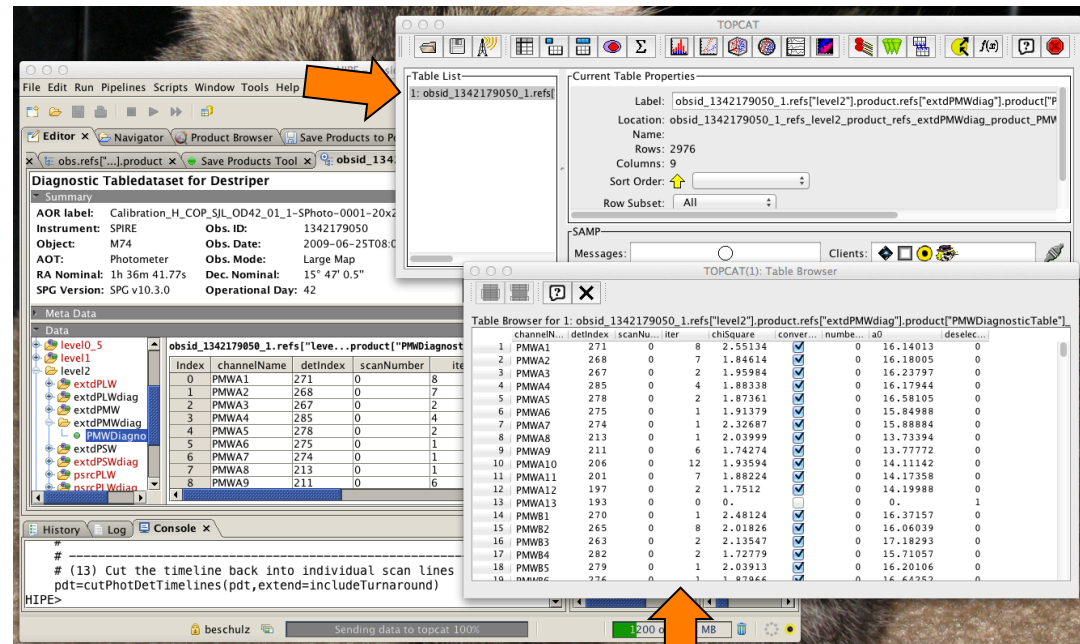
```
HIPE> del(obsid_1342179050_1_refs_level0_5_extdPMWdiag_product_PMWDiagnosticTable)
```

beschulz Jython Interpreter 100%



Data Exchange via SAMP (Tables)

- Another example sending a table dataset.
- Start Topcat on your computer.
- Select a SPIRE diagnostic table and send it to Topcat.
- It appears as a table and can be analyzed using the Topcat features.
- It can even be modified on Topcat and sent back to HIPE using the “Transmit table to all applications using SAMP” button.





Data Exchange via SAMP (Tables)

- Another example sending a table dataset.
- Start Topcat on your computer.
- Select a SPIRE diagnostic table and send it to Topcat.
- It appears as a table and can be analyzed using the Topcat features.
- It can even be modified on Topcat and sent back to HIPE using the “Transmit table to all applications using SAMP” button.



Note: The shown modification of a detector name is only for illustration and won't be useful otherwise .

The screenshot shows the Topcat software interface. A 'TableDataset' window displays a table with columns 'name', 'value', and 'unit'. Below it, a 'Data' window shows a table with columns 'Index', 'channelName', 'detIndex', 'scanNumber', and 'ite'. An orange arrow points to the row with 'channelName' 'PMWA5XXX'. A 'Table Browser for 1: obsid_1342179050_1.refsf' window is open, showing a list of channel names from PMWA1 to PMWA13, PMWB1 to PMWB5. An orange arrow points to 'PMWA5XXX' in this list. A 'Table List' window is also visible, showing '1: obsid_1342179050_1.refsf'. The bottom status bar indicates 'Receiving data from topcat 100%'.



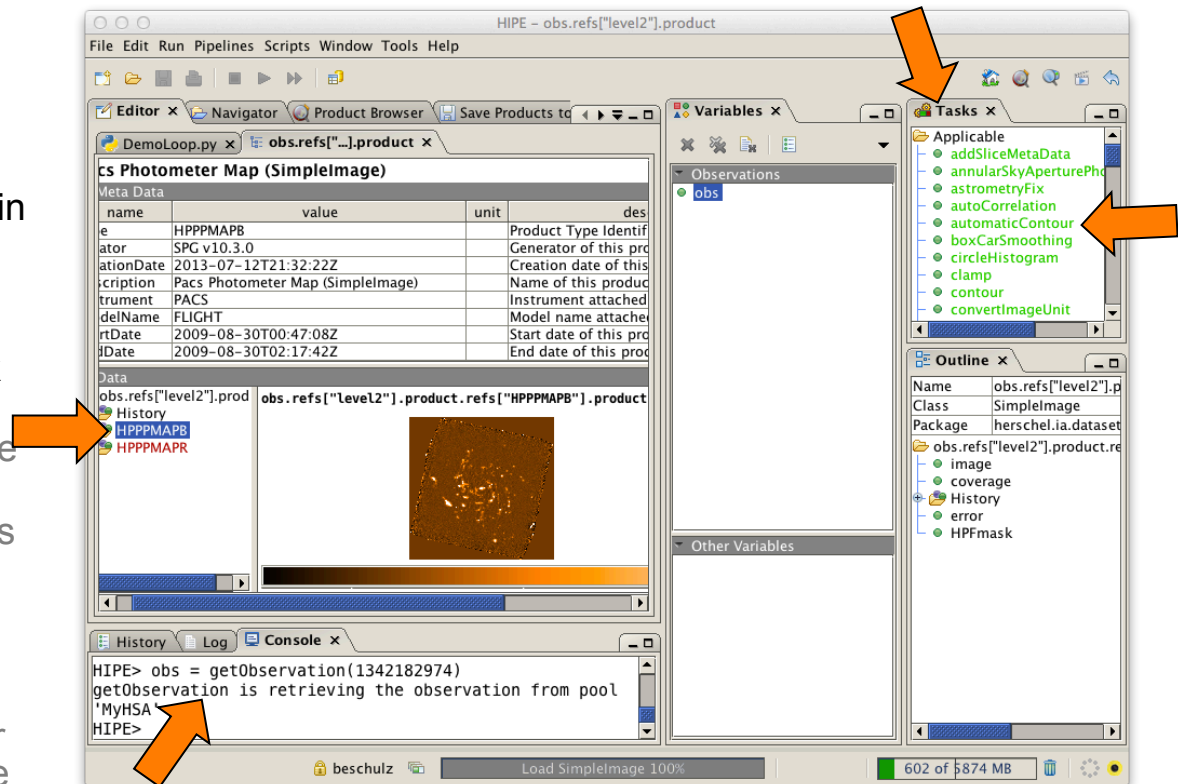
Tasks

- Tasks are special classes in HIPE conforming to a **rigorous specification of input and output parameters**.
- They are registered with the system and appear in the Task View.
- Calling them from the Task View will produce a default GUI regardless of whether GUI features were specified in the code.
- They can be made applicable to specific product types and will appear in the “Applicable” folder only for these.
- There is a large collection of pre-defined tasks available in HIPE.
- Users can write their own tasks in Jython if needed and distribute them as a HIPE Plug-In.
- SPIA and CASSIS are examples for HIPE Plug-Ins.



Task Example

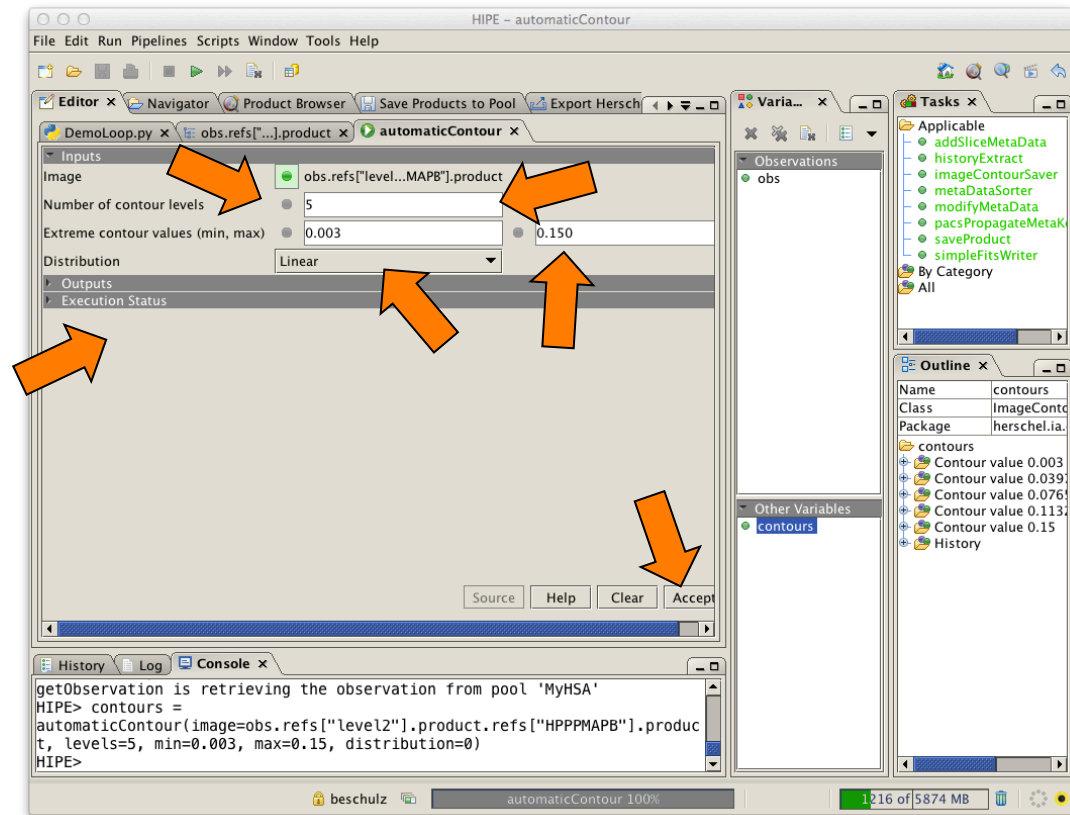
- Tasks appear in the Tasks View on the upper right in this perspective.
- Load a PACS observation, click “Level 2” in the Outline View and select HPPBMAPB in the appearing viewer.
- Double-click the “Applicable” folder in the Task View.
- Find and double-click the task “automaticContour”.
- The task GUI will appear in the Editor View.
- Enter some appropriate values and hit the “Accept” button.
- This will run the task.
- Note the script output in the Console View.
- Go back to the Level 2 Viewer and display the image with the Standard Image Viewer (right click).
- Drag the variable “contours” over the image.





Task Example

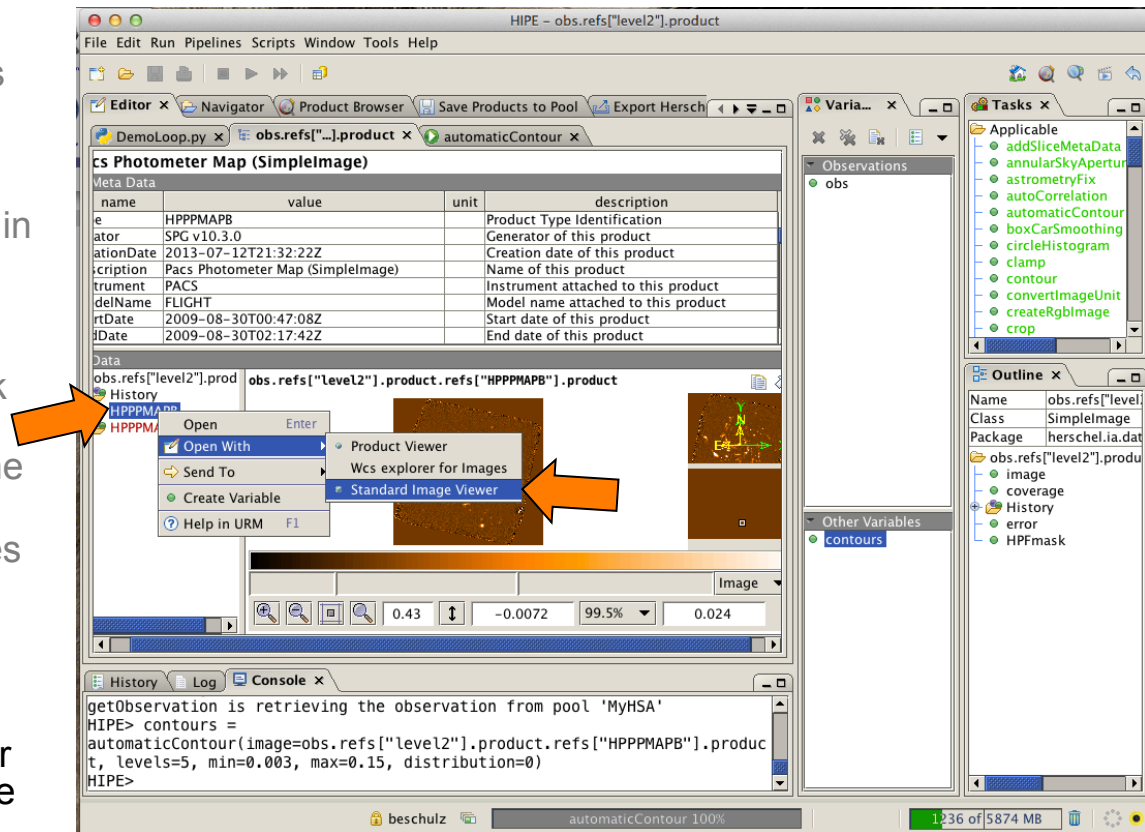
- Tasks appear in the Tasks View on the upper right in this perspective.
- Load a PACS observation, click “Level 2” in the Outline View and select HPPBMAPB in the appearing viewer.
- Double-click the “Applicable” folder in the Task View.
- Find and double-click the task “automaticContour”.
- The task GUI will appear in the Editor View.
- Enter some appropriate values and hit the “Accept” button.
- This will run the task.
- Note the script output in the Console View.
- Go back to the Level 2 Viewer and display the image with the Standard Image Viewer (right click).
- Drag the variable “contours” over the image.





Task Example

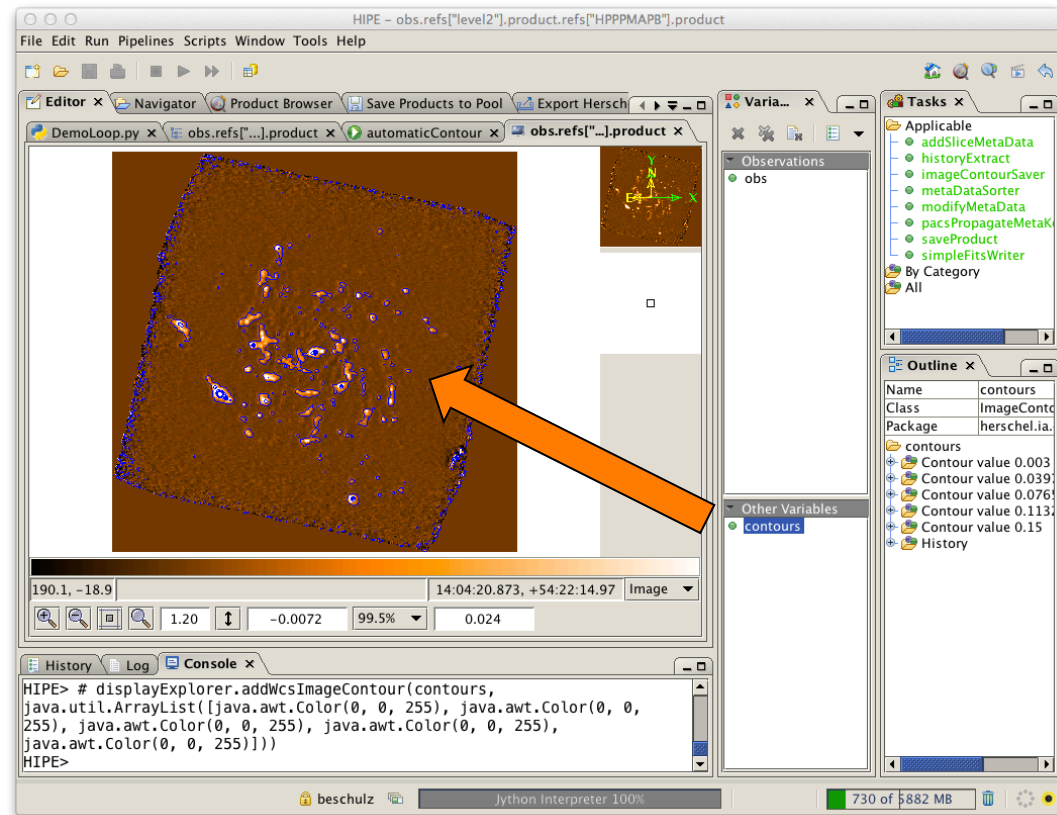
- Tasks appear in the Tasks View on the upper right in this perspective.
- Load a PACS observation, click “Level 2” in the Outline View and select HPPBMAPB in the appearing viewer.
- Double-click the “Applicable” folder in the Task View.
- Find and double-click the task “automaticContour”.
- The task GUI will appear in the Editor View.
- Enter some appropriate values and hit the “Accept” button.
- This will run the task.
- Note the script output in the Console View.
- Go back to the Level 2 Viewer and display the image with the Standard Image Viewer (right click).
- Drag the variable “contours” over the image.





Task Example

- Tasks appear in the Tasks View on the upper right in this perspective.
- Load a PACS observation, click “Level 2” in the Outline View and select HPPBMAPB in the appearing viewer.
- Double-click the “Applicable” folder in the Task View.
- Find and double-click the task “automaticContour”.
- The task GUI will appear in the Editor View.
- Enter some appropriate values and hit the “Accept” button.
- This will run the task.
- Note the script output in the Console View.
- Go back to the Level 2 Viewer and display the image with the Standard Image Viewer (right click).
- Drag the variable “contours” over the image.



Many more tasks are available, in particular for map arithmetic, photometry, line fitting etc.



Libraries/Packages

- HiPE contains a number of software Libraries that are available to scripting.
- Examples are:
 - Numerics Library
 - Plot Library
 - Image toolbox
 - Spectral toolbox
 - Product Access Layer
 - etc...
- A good overview over available functionality is in the “Categorized view of commands” in the “HCSS User’s Reference Manual”



Plot Library

- HIPE contains a powerful plot library with many ways to create high quality publication ready diagrams.
- A simple example code is shown to the right and the resulting plot.
- For more information refer to the documentation in the Data Analysis Guide.

The screenshot displays the HIPE software interface. The main window shows a Python script in the Editor pane:

```
1 x = Double1d.range(11) # Creates array with values from 0.0 to 10.0
2 y = x*x
3 myPlot = PlotXY()
4 myLayer = LayerXY(x,y)
5 myPlot.addLayer(myLayer)
6 myPlot.width = 400
7 myPlot.height = 300
8 myPlot.titleText = "Example plot"
9 myPlot.subtitleText = "Example subtitle"
```

The Console pane at the bottom shows the execution of the script:

```
HIPE> myPlot.width = 400
HIPE> myPlot.height = 300
HIPE> myPlot.titleText = "Example plot"
HIPE> myPlot.subtitleText = "Example subtitle"
HIPE>
```

An inset window titled "Herschel PlotXY" displays the resulting plot. The plot has a title "Example plot" and a subtitle "Example subtitle". The x-axis is labeled "x axis" and ranges from -1 to 11. The y-axis is labeled "y axis" and ranges from -10 to 110. A blue curve is plotted, representing the function $y = x^2$.



Plot Package

- HIPE contains a powerful plot library with many ways to create high quality publication ready diagrams.
- A simple example code is shown to the right and the resulting plot.
- For more information refer to the documentation in the Data Analysis Guide.

The screenshot shows the HIPE software interface. On the left is a Table of Contents (TOC) pane with sections: Introductory, Analysis Tools, HIFI, SPIRE, PACS, and Reference. The 'Analysis Tools' section is expanded to show '3. Plotting'. On the right is the main content area displaying 'Chapter 3. Plotting' with a 'Table of Contents' list of 26 numbered sub-sections, including 'Getting started', 'Creating a plot', 'Customising title and subtitle', 'Managing layers', 'Showing and customising a legend', 'Customising plot properties', 'Setting margins', 'Saving and printing', 'Setting line and symbol styles', 'Customising axes', 'Drawing grid lines', 'Managing annotations', 'Drawing filled areas', 'Drawing a horizontal or vertical line', 'Customising auxiliary axes', 'Changing the thickness of axes', 'Adding error bars', 'Switching to histogram mode', 'Adding subplots', 'Embedding monochromatic images in plots', 'Embedding RGB images in plots', 'Inserting math and special symbols', 'Creating a plot in batch mode', 'Drawing multiple plots per window', 'Colours in plots', and 'Methods for colours, fonts and visibility'.



Some Practical Points at the End

- This was only the tip of the Iceberg. HIPE is a powerful data processing and analysis system with **many more aspects to explore**.
- **Configuring HIPE** is important at the start.
- Set the places for the local pools (localStore) and for the HSA cache pool (myHsa) to a disk with **enough space**.
- The amount of RAM to reserve for HIPE should be adjusted to the RAM available and should **leave some room** for the operating system.
- See the **Preferences menu** for other adjustments, like the appearance of the console. In many cases you will have to re-start the session, so configure first.
- **Instruments** will have additional configuration requirements.
- There is **more documentation available** than you might wish for but we are here to give you the necessary guidance.